# BBC Micro OS 1.20
by Acorn Computers

(Partial) Annotated dis-assembly

By Kevin Edwards

From 1983 → 86.

Scanned October 2019

Twitter @KevEdwardsRetro

Please Fill in the gaps yourself!

Enjoy.

KE

OS 1.20

Character definitions.

```
?T C000 C300
C000 00 00 00 00 00 00 00 00  ........  Space
C008 18 18 18 18 18 00 18 00  ........  !
C010 6C 6C 6C 00 00 00 00 00  lll.....
C018 36 36 7F 36 7F 36 36 00  66.6.66.
C020 0C 3F 68 3E 0B 7E 18 00  .?h>.~..
C028 60 66 0C 18 30 66 06 00  `f..0f..
C030 38 6C 6C 38 6D 66 3B 00  8ll8mf;.
C038 0C 18 30 00 00 00 00 00  ..0.....
C040 0C 18 30 30 30 18 0C 00  ..000...
C048 30 18 0C 0C 0C 18 30 00  0.....0.
C050 00 18 7E 3C 7E 18 00 00  ..~<~...
C058 00 18 18 7E 18 18 00 00  ...~....
C060 00 00 00 00 00 18 18 30  .......0
C068 00 00 00 7E 00 00 00 00  ...~....
C070 00 00 00 00 00 18 18 00  ........
C078 00 06 0C 18 30 60 00 00  ....0`..
C080 3C 66 6E 7E 76 66 3C 00  <fn~vf<.
C088 18 38 18 18 18 18 7E 00  .8....~.
C090 3C 66 06 0C 18 30 7E 00  <f...0~.
C098 3C 66 06 1C 06 66 3C 00  <f...f<.
C0A0 0C 1C 3C 6C 7E 0C 0C 00  ..<l~...
C0A8 7E 60 7C 06 06 66 3C 00  ~`|..f<.
C0B0 1C 30 60 7C 66 66 3C 00  .0`|ff<.
C0B8 7E 06 0C 18 30 30 30 00  ~...000.
C0C0 3C 66 66 3C 66 66 3C 00  <ff<ff<.
C0C8 3C 66 66 3E 06 0C 38 00  <ff>..8.
C0D0 00 00 18 18 00 18 18 00  ........
C0D8 00 00 18 18 00 18 18 30  .......0
C0E0 0C 18 30 60 30 18 0C 00  ..0`0...
C0E8 00 00 7E 00 7E 00 00 00  ..~.~...
C0F0 30 18 0C 06 0C 18 30 00  0.....0.
C0F8 3C 66 0C 18 18 00 18 00  <f......
C100 3C 66 6E 6A 6E 60 3C 00  <fnjn`<.
C108 3C 66 66 7E 66 66 66 00  <ff~fff.
C110 7C 66 66 7C 66 66 7C 00  |ff|ff|.
C118 3C 66 60 60 60 66 3C 00  <f```f<.
C120 78 6C 66 66 66 6C 78 00  xlfffl x.
C128 7E 60 60 7C 60 60 7E 00  ~``|``~.
C130 7E 60 60 7C 60 60 60 00  ~``|```.
C138 3C 66 60 6E 66 66 3C 00  <f`nff<.
C140 66 66 66 7E 66 66 66 00  fff~fff.
C148 7E 18 18 18 18 18 7E 00  ~.....~.
C150 3E 0C 0C 0C 0C 6C 38 00  >....l8.
C158 66 6C 78 70 78 6C 66 00  flxpxlf.
C160 60 60 60 60 60 60 7E 00  ``````~.
C168 63 77 7F 6B 6B 63 63 00  cw.kkcc.
C170 66 66 76 7E 6E 66 66 00  ffv~nff.
C178 3C 66 66 66 66 66 3C 00  <fffff<.
C180 7C 66 66 7C 60 60 60 00  |ff|```.
C188 3C 66 66 66 6A 6C 36 00  <fffjl6.
C190 7C 66 66 7C 6C 66 66 00  |ff|lff.
C198 3C 66 60 3C 06 66 3C 00  <f`<.f<.
C1A0 7E 18 18 18 18 18 18 00  ~.......
C1A8 66 66 66 66 66 66 3C 00  ffffff<.
C1B0 66 66 66 66 66 3C 18 00  fffff<..
C1B8 63 63 6B 6B 7F 77 63 00  cckk.wc.
C1C0 66 66 3C 18 3C 66 66 00  ff<.<ff.
C1C8 66 66 66 3C 18 18 18 00  fff<....
C1D0 7E 06 0C 18 30 60 7E 00  ~...0`~.
C1D8 7C 60 60 60 60 60 7C 00  |`````|.
```

All Data

```
C1E0 00 60 30 18 0C 06 00 00   .`0.....
C1E8 3E 06 06 06 06 06 3E 00   >.....>.
C1F0 18 3C 66 42 00 00 00 00   .<fB....
C1F8 00 00 00 00 00 00 00 FF   ........
C200 1C 36 30 7C 30 30 7E 00   .60|00~.
C208 00 00 3C 06 3E 66 3E 00   ..<.>f>.
C210 60 60 7C 66 66 66 7C 00   ``|fff|.
C218 00 00 3C 66 60 66 3C 00   ..<f`f<.
C220 06 06 3E 66 66 66 3E 00   ..>fff>.
C228 00 00 3C 66 7E 60 3C 00   ..<f~`<.
C230 1C 30 30 7C 30 30 30 00   .00|000.
C238 00 00 3E 66 66 3E 06 3C   ..>ff>.<
C240 60 60 7C 66 66 66 66 00   ``|ffff.
C248 18 00 38 18 18 18 3C 00   ..8...<.
C250 18 00 38 18 18 18 18 70   ..8....p
C258 60 60 66 6C 78 6C 66 00   ``flxlf.
C260 38 18 18 18 18 18 3C 00   8.....<.
C268 00 00 36 7F 6B 6B 63 00   ..6.kkc.
C270 00 00 7C 66 66 66 66 00   ..|ffff.
C278 00 00 3C 66 66 66 3C 00   ..<fff<.
C280 00 00 7C 66 66 7C 60 60   ..|ff|``
C288 00 00 3E 66 66 3E 06 07   ..>ff>..
C290 00 00 6C 76 60 60 60 00   ..lv```.
C298 00 00 3E 60 3C 06 7C 00   ..>`<.|.
C2A0 30 30 7C 30 30 30 1C 00   00|000..
C2A8 00 00 66 66 66 66 3E 00   ..ffff>.
C2B0 00 00 66 66 66 3C 18 00   ..fff<..
C2B8 00 00 63 6B 6B 7F 36 00   ..ckk.6.
C2C0 00 00 66 3C 18 3C 66 00   ..f<.<f.
C2C8 00 00 66 66 66 3E 06 3C   ..fff>.<
C2D0 00 00 7E 0C 18 30 7E 00   ..~..0~.
C2D8 0C 18 18 70 18 18 0C 00   ...p....
C2E0 18 18 18 00 18 18 18 00   ........
C2E8 30 18 18 0E 18 18 30 00   0.....0.
C2F0 31 6B 46 00 00 00 00 00   1kF.....
C2F8 FF FF FF FF FF FF FF FF   ........

?0FF
```

All Data

```
?D C300 F000
C300 4C 1D CB     L..     JMP  &CB1D
C303 0D 42 42     .BB     ORA  &4242
C306 43           C       ???
C307 20 43 6F      Co     JSR  &6F43
C30A 6D 70 75     mpu     ADC  &7570
C30D 74           t       ???
C30E 65 72        er      ADC  &72
C310 20 00 31      .1     JSR  &3100
C313 36 4B        6K      ROL  &4B,X
C315 07           .       ???
C316 00           .       BRK
C317 33           3       ???
C318 32           2       ???
C319 4B           K       ???
C31A 07           .       ???
C31B 00           .       BRK
C31C 08           .       PHP
C31D 0D 0D 00     ...     ORA  &000D
C320 11 22        ."      ORA  (&22),Y
C322 33           3       ???
C323 44           D       ???
C324 55 66        Uf      EOR  &66,X
C326 77           w       ???
C327 88           .       DEY
C328 99 AA BB     ...     STA  &BBAA,Y
C32B CC DD EE     ...     CPY  &EEDD
C32E FF           .       ???
C32F 00           .       BRK
C330 55 AA        U.      EOR  &AA,X
C332 FF           .       ???
C333 11 3B        .;      ORA  (&3B),Y
C335 96 A1        ..      STX  &A1,Y
C337 AD B9 11     ...     LDA  &11B9
C33A 6F           o       ???
C33B C5 64        .d      CMP  &64
C33D F0 5B        .[      BEQ  &C39A
C33F 59 AF 8D     Y..     EOR  &8DAF,Y
C342 A6 C0        ..      LDX  &C0
C344 F9 FD 92     ...     SBC  &92FD,Y
C347 39 9B EB     9..     AND  &EB9B,Y
C34A F1 39        .9      SBC  (&39),Y
C34C 8C BD 11     ...     STY  &11BD
C34F FA           .       ???
C350 A2 79        .y      LDX  #&79
C352 87           .       ???
C353 AC C5 2F     ../     LDY  &2FC5
C356 C5 C5        ..      CMP  &C5
C358 C5 C5        ..      CMP  &C5
C35A C5 E8        ..      CMP  &E8
C35C C5 C6        ..      CMP  &C6
C35E C6 C6        ..      DEC  &C6
C360 C7           .       ???
C361 C7           .       ???
C362 C5 C5        ..      CMP  &C5
C364 C7           .       ???
C365 4F           O       ???
C366 4E 5B C8     N[.     LSR  &C85B
C369 C5 5F        ._      CMP  &5F
C36B 57           W       ???
C36C 78           x       SEI
```

Mode change routine.

'BBC Computer 32K' (16K)

+ Beep if CTRL + BREAK

+ A few CR's.

Data

byte masks for 4 colour Modes.

byte masks for 16 colour Mode.

Addresses for decoded VDU codes Ø-31

low bytes of action

Addresses for VDU's (Ø - 32) ? 32.

NB (32 is the delete function 127 see &C4D2.

high bytes of Address

-ve = true address if it has no extra high byte of parameters,

~~+ve = Address &C3~~

if it requires more parame

```
C36D 6B          k       ???
C36E C9 C5       ..      CMP   #&C5
C370 3C          <       ???
C371 7C          |       ???
C372 C7          .       ???
C373 4E CA 00    N..     LSR   &00CA
C376 00          .       BRK
C377 02          .       ???
C378 80          .       ???
C379 05 00       ..      ORA   &00
C37B 07          .       ???
C37C 80          .       ???
C37D 0A          .       ASL   A
C37E 00          .       BRK
C37F 0C          .       ???
C380 80          .       ???
C381 0F          .       ???
C382 00          .       BRK
C383 11 80       ..      ORA   (&80),Y
C385 14          .       ???
C386 00          .       BRK
C387 16 80       ..      ASL   &80,X
C389 19 00 1B    ...     ORA   &1B00,Y
C38C 80          .       ???
C38D 1E 00 20    ..      ASL   &2000,X
C390 80          .       ???
C391 23          #       ???
C392 00          .       BRK
C393 25 80       %.      AND   &80
C395 28          (       PLP
C396 00          .       BRK
C397 2A          *       ROL   A
C398 80          .       ???
C399 2D 00 2F    -./     AND   &2F00
C39C 80          .       ???
C39D 32          2       ???
C39E 00          .       BRK
C39F 34          4       ???
C3A0 80          .       ???
C3A1 37          7       ???
C3A2 00          .       BRK
C3A3 39 80 3C    9.<     AND   &3C80,Y
C3A6 00          .       BRK
C3A7 3E 80 41    >.A     ROL   &4180,X
C3AA 00          .       BRK
C3AB 43          C       ???
C3AC 80          .       ???
C3AD 46 00       F.      LSR   &00
C3AF 48          H       PHA
C3B0 80          .       ???
C3B1 4B          K       ???
C3B2 00          .       BRK
C3B3 4D 80 00    M..     EOR   &0080
C3B6 00          .       BRK
C3B7 00          .       BRK
C3B8 28          (       PLP
C3B9 00          .       BRK
C3BA 50 00       P.      BVC   &C3BC
C3BC 78          x       SEI
C3BD 00          .       BRK
C3BE A0 00       ..      LDY   #&00
C3C0 C8          .       INY
```

cont.
high bytes of VDU's
cont.

Location 4

32 entries of a 640* Multiplication table

All Data.

25 entries of a 40 * Multiplication table

```
C3C1 00          .      BRK
C3C2 F0 01       ..     BEQ  &C3C5
C3C4 18          .      CLC
C3C5 01 40       .@     ORA  (&40,X)
C3C7 01 68       .h     ORA  (&68,X)
C3C9 01 90       ..     ORA  (&90,X)
C3CB 01 B8       ..     ORA  (&B8,X)
C3CD 01 E0       ..     ORA  (&E0,X)
C3CF 02          .      ???
C3D0 08          .      PHP
C3D1 02          .      ???
C3D2 30 02       0.     BMI  &C3D6
C3D4 58          X      CLI
C3D5 02          .      ???
C3D6 80          .      ???
C3D7 02          .      ???
C3D8 A8          .      TAY
C3D9 02          .      ???
C3DA D0 02       ..     BNE  &C3DE
C3DC F8          .      SED
C3DD 03          .      ???
C3DE 20 03 48     .H    JSR  &4803
C3E1 03          .      ???
C3E2 70 03       p.     BVS  &C3E7
C3E4 98          .      TYA
C3E5 03          .      ???
C3E6 C0 1F       ..     CPY  #&1F
C3E8 1F          .      ???
C3E9 1F          .      ???
C3EA 18          .      CLC
C3EB 1F          .      ???
. CLC
C3EE 18          .      CLC
C3EF 4F          O      ???
C3F0 27          '      ???
C3F1 13          .      ???
C3F2 4F          O      ???
C3F3 27          '      ???
C3F4 13          .      ???
C3F5 27          '      ???
C3F6 27          '      ???
C3F7 9C          .      ???
C3F8 D8          .      CLD
C3F9 F4          .      ???
C3FA 9C          .      ???
C3FB 88          .      DEY
C3FC C4 88       ..     CPY  &88
C3FE 4B          K      ???
C3FF 08          .      PHP
C400 10 20       .      BPL  &C422
C402 08          .      PHP
C403 08          .      PHP
C404 10 08       ..     BPL  &C40E
C406 01 AA       ..     ORA  (&AA,X)
C408 55 88       U.     EOR  &88,X
C40A 44          D      ???
C40B 22          "      ???
C40C 11 80       ..     ORA  (&80),Y
C40E 40          @      RTI
C40F 20 10 08     ..    JSR  &0810
C412 04          .      ???
C413 02          .      ???
```

Cont.

5

All Data

No of character rows − 1 For all Modes.

No of character columns − 1 For all Modes.

Video ULA codes For all Modes

No of bytes storage taken per character.

Mask table For 16 colour Mode (2)

Mask table For 4 colour Modes.

Mask table For 2 colour Modes.

```
C414 01 03        ..      ORA   (&03,X)
C416 0F           .       ???
C417 01 01        ..      ORA   (&01,X)
C419 03           .       ???
C41A 01 00        ..      ORA   (&00,X)
C41C FF           .       ???
C41D 00           .       BRK
C41E 00           .       BRK
C41F FF           .       ???
C420 FF           .       ???
C421 FF           .       ???
C422 FF           .       ???
C423 00           .       BRK
C424 00           .       BRK
C425 FF           .       ???
C426 00           .       BRK
C427 0F           .       ???
C428 F0 FF        ..      BEQ   &C429
C42A 00           .       BRK
C42B 03           .       ???
C42C 0C           .       ???
C42D 0F           .       ???
C42E 30 33        03      BMI   &C463
C430 3C           <       ???
C431 3F           ?       ???
C432 C0 C3        ..      CPY   #&C3
C434 CC CF F0     ...     CPY   &F0CF
C437 F3           .       ???
C438 FC           .       ???
C439 FF           .       ???
C43A 07           .       ???
 ..       ORA   (&00,X)
C43E 07           .       ???
C43F 03           .       ???
C440 00           .       BRK
C441 00           .       BRK
C442 00           .       BRK
C443 01 02        ..      ORA   (&02,X)
C445 02           .       ???
C446 03           .       ???
C447 04           .       ???
C448 00           .       BRK
C449 06 02        ..      ASL   &02
C44B 0D 05 0D     ...     ORA   &0D05
C44E 05 04        ..      ORA   &04
C450 04           .       ???
C451 0C           .       ???
C452 0C           .       ???
C453 04           .       ???
C454 02           .       ???
C455 32           2       ???
C456 7A           z       ???
C457 92           .       ???
C458 E6 50        .P      INC   &50
C45A 40           @       RTI
C45B 28           (       PLP
C45C 20 04 30      .0     JSR   &3004
C45F 40           @       RTI
C460 58           X       CLI
C461 60           `       RTS
C462 7C           !       ???
C463 28           (       PLP
```

see &CBDA

cont.

No of colours − 1 For each Mode.

4 x 5 byte tables used to process the GCOL plotting options.

4 col table

Data.

16 col table.

No of pixels per byte − 1 For each Mode

Memory Map type For all the modes. 0=20K 2=16K 4=1K 1=16K 3=8K

Various VDU control numbers.

VIA data For Screen Size

DATA 2 For Scrolling

two bytes For vector For CLS in different Mode

MSB of no of bytes taken up by screen. (0-4)

LSB " " " " " " " " (0-4)

PTO

```
C464 40          @       RTI
C465 80          .       ???
C466 B5 75       .u      LDA   &75,X
C468 75 0B       u.      ADC   &0B,X
C46A 17          .       ???
C46B 23          #       ???
C46C 2F          /       ???
C46D 3B          ;       ???
C46E 7F          .       ???
C46F 50 62       Pb      BVC   &C4D3
C471 28          (       PLP
C472 26 00       &.      ROL   &00
C474 20 22 01     ".     JSR   &0122
C477 07          .       ???
C478 67          g       ???
C479 08          .       PHP
C47A 7F          .       ???
C47B 50 62       Pb      BVC   &C4DF
C47D 28          (       PLP
C47E 1E 02 19    ...     ASL   &1902,X
C481 1B          .       ???
C482 01 09       ..      ORA   (&09,X)
C484 67          g       ???
C485 09 3F       .?      ORA   #&3F
C487 28          (       PLP
C488 31 24       1$      AND   (&24),Y
C48A 26 00       &.      ROL   &00
C48C 20 22 01     ".     JSR   &0122
C48F 07          .       ???
C490 67          g       ???
C491 08          .       PHP
C492 3F          ?       ???
C493 28          (       PLP
C494 31 24       1$      AND   (&24),Y
C496 1E 02 19    ...     ASL   &1902,X
C499 1B          .       ???
C49A 01 09       ..      ORA   (&09,X)
C49C 67          g       ???
C49D 09 3F       .?      ORA   #&3F
C49F 28          (       PLP
C4A0 33          3       ???
C4A1 24 1E       $.      BIT   &1E
C4A3 02          .       ???
C4A4 19 1B 93    ...     ORA   &931B,Y
C4A7 12          .       ???
C4A8 72          r       ???
C4A9 13          .       ???
C4AA 86 D3       ..      STX   &D3
C4AC 7E D3 6A    ~.j     ROR   &6AD3,X
C4AF 74          t       ???
C4B0 42          B       ???
C4B1 4B          K       ???
C4B2 D3          .       ???
C4B3 D3          .       ???
C4B4 D3          .       ???
C4B5 D3          .       ???
C4B6 23          #       ???
C4B7 5F          _       ???
C4B8 60          `       RTS
C4B9 23          #       ???
C4BA 04          .       ???
C4BB 05 06       ..      ORA   &06
```

640* table low +&C3XX

&C46E offset for 6845

tables used by the VDU section to index into other tables.

6845 reg 0-11 For Modes 0-2.

6845 reg 0-11 For Mode 3.

6845 reg 0-11 For Modes 4 & 5.

All Data

6845 reg 0-11 For Mode 6.

6845 reg 0-11 For Mode 7.

VDU Driver JUMP table

See &D220.

teletext conversion table. Data

From to

fontexplosion &0.

See & CD24

# Main VDU Section

```
C4BD 00          .      BRK
C4BE 01 02       ..     ORA   (&02,X)
C4C0 AE 6A 02    .j.    LDX   &026A     no of items in VDU queue.
C4C3 D0 4D       .M     BNE   &C512     branch if not empty.
C4C5 24 D0       $.     BIT   &D0       read VDU status. (P134 AUG)
C4C7 50 0F       P.     BVC   &C4D8     branch if cursors are together
C4C9 20 68 C5     h.    JSR   &C568     Move cursor (6845) to new posn
C4CC 20 6A CD     j.    JSR   &CD6A     Display Software cursor.
C4CF 30 07       0.     BMI   &C4D8     (&DØ from &CACS) branch if VDU21 is.
C4D1 C9 0D       ..     CMP   #&0D      } Branch if CR being printed
C4D3 D0 03       ..     BNE   &C4D8     }
C4D5 20 18 D9     ..    JSR   &D918     Mix cursor operations.
C4D8 C9 7F       ..     CMP   #&7F      } is it 'delete'
C4DA F0 11       ..     BEQ   &C4ED     } branch if true.
C4DC C9 20       .      CMP   #&20      } is it ~~Space~~ below a Space character
C4DE 90 0F       ..     BCC   &C4EF     } branch if character <32 (&2Ø).
C4E0 24 D0       $.     BIT   &D0       VDU Status. (P134 AUG)
C4E2 30 06       0.     BMI   &C4EA     branch if VDU disabled (by VDU21)
C4E4 20 B7 CF     ..    JSR   &CFB7     display character.
C4E7 20 64 C6     d.    JSR   &C664     Move text cursor (after 6845).
C4EA 4C 5E C5    L^.    JMP   &C55E
C4ED A9 20       .      LDA   #&20
C4EF A8          .      TAY             Y=character being printed.
C4F0 B9 33 C3    .3.    LDA   &C333,Y   - VDU decoding table (low bytes)?
C4F3 8D 5D 03    .].    STA   &035D     - general jump vector
C4F6 B9 54 C3    .T.    LDA   &C354,Y   - VDU decoding table (high bytes)?
C4F9 30 4A       0J     BMI   &C545     - in general, this calls the routine.
C4FB AA          .      TAX             X= value from high of address.
C4FC 09 F0       ..     ORA   #&F0      set top 4 bits.
C4FE 8D 6A 02    .j.    STA   &026A     no of items in VDU queue.
C501 8A          .      TXA             retrieve byte from table.
C502 4A          J      LSR   A
C503 4A          J      LSR   A         Move upper nibble to lower nibble.
C504 4A          J      LSR   A
C505 4A          J      LSR   A
C506 18          .      CLC             } Add &C3 to high byte of (address /16)
C507 69 C3       i.     ADC   #&C3      }
C509 8D 5E 03    .^.    STA   &035E     Save in VDU vector high byte.
C50C 24 D0       $.     BIT   &D0       check VDU status.
C50E 70 1F       p.     BVS   &C52F     if bit 6 is set branch (if cursors are separated).
C510 18          .      CLC             C=Ø
C511 60          `      RTS             exit.
C512 9D 24 02    .$.    STA   &0224,X   Save char in VDU queue & inc pointer
C515 E8          .      INX
C516 8E 6A 02    .j.    STX   &026A
C519 D0 17       ..     BNE   &C532     Branch if VDU not complete
C51B 24 D0       $.     BIT   &D0
C51D 30 15       0.     BMI   &C534     Branch if VDU21 has disabled VDU driver
C51F 70 05       p.     BVS   &C526     Branch if cursors are separate.
C521 20 F5 CC     .     JSR   &CCF5     (JMP (&35D)!)
C524 18          .      CLC
C525 60          `      RTS
C526 20 68 C5     h.    JSR   &C568
C529 20 6A CD     j.    JSR   &CD6A
C52C 20 F5 CC     ..    JSR   &CCF5     JMP (&35D)
C52F 20 65 C5     e.    JSR   &C565
C532 18          .      CLC
C533 60          `      RTS
C534 AC 5E 03    .^.    LDY   &035E
C537 C0 C5       ..     CPY   #&C5
C539 D0 F7       ..     BNE   &C532
C53B AA          .      TAX
```

(2's comp of bytes required before leaving VDU.)

VDUØ, 6 VDU6 & VDU 27 | Multi byte VDU instructions.

```
C53C A5 D0        ..      LDA  &D0
C53E 4A           J       LSR  A
C53F 90 D0        ..      BCC  &C511
C541 8A           .       TXA
C542 4C 1E E1     L..     JMP  &E11E
C545 8D 5E 03     .^.     STA  &035E
C548 98           .       TYA
C549 C9 08        ..      CMP  #&08
C54B 90 06        ..      BCC  &C553
C54D 49 FF        I.      EOR  #&FF
C54F C9 F2        ..      CMP  #&F2
C551 49 FF        I.      EOR  #&FF
C553 24 D0        $.      BIT  &D0
C555 30 29        0)      BMI  &C580
C557 08           .       PHP
C558 20 F5 CC      ..     JSR  &CCF5
C55B 28           (       PLP
C55C 90 03        ..      BCC  &C561
C55E A5 D0        ..      LDA  &D0
C560 4A           J       LSR  A
C561 24 D0        $.      BIT  &D0
C563 50 AC        P.      BVC  &C511
C565 20 7A CD      z.     JSR  &CD7A
C568 08           .       PHP
C569 48           H       PHA
C56A A2 18        ..      LDX  #&18
C56C A0 64        .d      LDY  #&64
C56E 20 DE CD      ..     JSR  &CDDE
C571 20 06 CF      ..     JSR  &CF06
C574 20 02 CA      ..     JSR  &CA02
C577 A5 D0        ..      LDA  &D0
C579 49 02        I.      EOR  #&02
C57B 85 D0        ..      STA  &D0
C57D 68           h       PLA
C57E 28           (       PLP
C57F 60           `       RTS
C580 49 06        I.      EOR  #&06
C582 D0 08        ..      BNE  &C58C
C584 A9 7F        ..      LDA  #&7F
C586 90 20        .       BCC  &C5A8
C588 A5 D0        ..      LDA  &D0
C58A 29 20        )       AND  #&20
C58C 60           `       RTS
C58D A0 00        ..      LDY  #&00
C58F 8C 69 02     .i.     STY  &0269
C592 A9 04        ..      LDA  #&04
C594 D0 07        ..      BNE  &C59D
C596 20 A2 E1      ..     JSR  &E1A2
C599 A9 94        ..      LDA  #&94
C59B 49 95        I.      EOR  #&95
C59D 05 D0        ..      ORA  &D0
C59F D0 09        ..      BNE  &C5AA
C5A1 20 A2 E1      ..     JSR  &E1A2
C5A4 A9 0A        ..      LDA  #&0A
C5A6 49 F4        I.      EOR  #&F4
C5A8 25 D0        %.      AND  &D0
C5AA 85 D0        ..      STA  &D0
C5AC 60           `       RTS
C5AD AD 61 03     .a.     LDA  &0361
C5B0 F0 FA        ..      BEQ  &C5AC
C5B2 20 51 C9      Q.     JSR  &C951
C5B5 A9 DF        ..      LDA  #&DF
```

VDU high byte for jump vector.
Y= character being printed.
is character less than 8.
branch if true.
invert bits.
CMP with &F2
restore bits.
VDU Status
branch if VDU drivers disabled (VDU
save status.
Jump through (&35D, &35E)
restore status.
was A EOR &FF < &F2, if so branch.
VDU status.
/2
test bit 6, branch if cursors are together
save SR & Acc,
swap current text Cursor coordinates with text input cursors co-ordin
Calc Scn Address of text cursor (in &D8, &D9
Move cursor (both one) to new posn.
Toggle 'Scrolling disabled' bit.
Restore St, A & Exit

VDU 14

enable page mode (set bit 2 of VDU statu

VDU 2.

VDU 21

VDU 3

VDU 15

VDU 4

```
C5B7 D0 EF        ..      BNE  &C5A8
C5B9 AD 61 03     .a.     LDA  &0361
C5BC F0 EE        ..      BEQ  &C5AC
C5BE A9 20        .       LDA  #&20
C5C0 20 54 C9      T.     JSR  &C954
C5C3 D0 D8        ..      BNE  &C59D
C5C5 20 88 C5      ..     JSR  &C588
C5C8 D0 55        .U      BNE  &C61F
C5CA CE 18 03     ...     DEC  &0318
C5CD AE 18 03     ...     LDX  &0318
C5D0 EC 08 03     ...     CPX  &0308
C5D3 30 19        0.      BMI  &C5EE
C5D5 AD 4A 03     .J.     LDA  &034A
C5D8 38           8       SEC
C5D9 ED 4F 03     .O.     SBC  &034F
C5DC AA           .       TAX
C5DD AD 4B 03     .K.     LDA  &034B
C5E0 E9 00        ..      SBC  #&00
C5E2 CD 4E 03     .N.     CMP  &034E
C5E5 B0 03        ..      BCS  &C5EA
C5E7 6D 54 03     mT.     ADC  &0354
C5EA A8           .       TAY
C5EB 4C F6 C9     L..     JMP  &C9F6
C5EE AD 0A 03     ...     LDA  &030A
C5F1 8D 18 03     ...     STA  &0318
C5F4 CE 69 02     .i.     DEC  &0269
C5F7 10 03        ..      BPL  &C5FC
C5F9 EE 69 02     .i.     INC  &0269
C5FC AE 19 03     ...     LDX  &0319
C5FF EC 0B 03     ...     CPX  &030B
C602 F0 06        ..      BEQ  &C60A
C604 CE 19 03     ...     DEC  &0319
C607 4C AF C6     L..     JMP  &C6AF
C60A 18           .       CLC
C60B 20 3F CD      ?.     JSR  &CD3F
C60E A9 08        ..      LDA  #&08
C610 24 D0        $.      BIT  &D0
C612 D0 05        ..      BNE  &C619
C614 20 94 C9      ..     JSR  &C994
C617 D0 03        ..      BNE  &C61C
C619 20 A4 CD      ..     JSR  &CDA4
C61C 4C AC C6     L..     JMP  &C6AC
C61F A2 00        ..      LDX  #&00
C621 86 DB        ..      STX  &DB
C623 20 0D D1      ..     JSR  &D10D
C626 A6 DB        ..      LDX  &DB
C628 38           8       SEC
C629 BD 24 03     .$.     LDA  &0324,X
C62C E9 08        ..      SBC  #&08
C62E 9D 24 03     .$.     STA  &0324,X
C631 B0 03        ..      BCS  &C636
C633 DE 25 03     .%.     DEC  &0325,X
C636 A5 DA        ..      LDA  &DA
C638 D0 1E        ..      BNE  &C658
C63A 20 0D D1      ..     JSR  &D10D
C63D F0 19        ..      BEQ  &C658
C63F A6 DB        ..      LDX  &DB
C641 BD 04 03     ...     LDA  &0304,X
C644 E0 01        ..      CPX  #&01
C646 B0 02        ..      BCS  &C64A
C648 E9 06        ..      SBC  #&06
C64A 9D 24 03     .$.     STA  &0324,X
```

VDU 5

VDU 8

```
C64D BD 05 03     ...     LDA  &0305,X
C650 E9 00        ..      SBC  #&00
C652 9D 25 03     .%.     STA  &0325,X
C655 8A           .       TXA
C656 F0 08        ..      BEQ  &C660
C658 4C B8 D1     L..     JMP  &D1B8
C65B 20 88 C5      ..     JSR  &C588
C65E F0 94        ..      BEQ  &C5F4
C660 A2 02        ..      LDX  #&02
C662 D0 52        .R      BNE  &C6B6
C664 A5 D0        ..      LDA  &D0
C666 29 20        )       AND  #&20
C668 D0 4A        .J      BNE  &C6B4
C66A AE 18 03     ...     LDX  &0318
C66D EC 0A 03     ...     CPX  &030A
C670 B0 12        ..      BCS  &C684
C672 EE 18 03     ...     INC  &0318
C675 AD 4A 03     .J.     LDA  &034A
C678 6D 4F 03     mO.     ADC  &034F
C67B AA           .       TAX
C67C AD 4B 03     .K.     LDA  &034B
C67F 69 00        i.      ADC  #&00
C681 4C F6 C9     L..     JMP  &C9F6
C684 AD 08 03     ...     LDA  &0308
C687 8D 18 03     ...     STA  &0318
C68A 18           .       CLC
C68B 20 E3 CA      ..     JSR  &CAE3
C68E AE 19 03     ...     LDX  &0319
C691 EC 09 03     ...     CPX  &0309
C694 B0 05        ..      BCS  &C69B
C696 EE 19 03     ...     INC  &0319
C699 90 14        ..      BCC  &C6AF
C69B 20 3F CD      ?.     JSR  &CD3F
C69E A9 08        ..      LDA  #&08
C6A0 24 D0        $.      BIT  &D0
C6A2 D0 05        ..      BNE  &C6A9
C6A4 20 A4 C9      ..     JSR  &C9A4
C6A7 D0 03        ..      BNE  &C6AC
C6A9 20 FF CD      ..     JSR  &CDFF
C6AC 20 AC CE      ..     JSR  &CEAC
C6AF 20 06 CF      ..     JSR  &CF06
C6B2 90 7E        .~      BCC  &C732
C6B4 A2 00        ..      LDX  #&00
C6B6 86 DB        ..      STX  &DB
C6B8 20 0D D1      ..     JSR  &D10D
C6BB A6 DB        ..      LDX  &DB
C6BD 18           .       CLC
C6BE BD 24 03     .$.     LDA  &0324,X
C6C1 69 08        i.      ADC  #&08
C6C3 9D 24 03     .$.     STA  &0324,X
C6C6 90 03        ..      BCC  &C6CB
C6C8 FE 25 03     .%.     INC  &0325,X
C6CB A5 DA        ..      LDA  &DA
C6CD D0 89        ..      BNE  &C658
C6CF 20 0D D1      ..     JSR  &D10D
C6D2 F0 84        ..      BEQ  &C658
C6D4 A6 DB        ..      LDX  &DB
C6D6 BD 00 03     ...     LDA  &0300,X
C6D9 E0 01        ..      CPX  #&01
C6DB 90 02        ..      BCC  &C6DF
C6DD 69 06        i.      ADC  #&06
C6DF 9D 24 03     .$.     STA  &0324,X
```

VDU 11

VDU 9

VDU status.
test VDU 5 bit
branch if VDU 5 enabled.
text cursor X co-ordinate.
right hand bottom column of text window
is equal or > branch.
increment X co-ordinate.
text cursor position (as sent to 6845) L
Add no of bytes taken up by 1 character
X = new value.
cell
text cursor high (as sent to 6845).
Add 0, or 1 if carry resulted.

```
C6E2 BD 01 03     ...     LDA  &0301,X
C6E5 69 00        i.      ADC  #&00
C6E7 9D 25 03     .%.     STA  &0325,X
C6EA 8A           .       TXA
C6EB F0 08        ..      BEQ  &C6F5
C6ED 4C B8 D1     L..     JMP  &D1B8
```

VDU 10

```
C6F0 20 88 C5      ..     JSR  &C588
C6F3 F0 95        ..      BEQ  &C68A
C6F5 A2 02        ..      LDX  #&02
C6F7 4C 21 C6     L!.     JMP  &C621
C6FA AE 55 03     .U.     LDX  &0355
C6FD AD 21 03     .!.     LDA  &0321
C700 CD 23 03     .#.     CMP  &0323
C703 90 53        .S      BCC  &C758
C705 DD E7 C3     ...     CMP  &C3E7,X
C708 F0 02        ..      BEQ  &C70C
C70A B0 4C        .L      BCS  &C758
C70C AD 22 03     ."."    LDA  &0322
C70F A8           .       TAY
C710 DD EF C3     ...     CMP  &C3EF,X
C713 F0 02        ..      BEQ  &C717
C715 B0 41        .A      BCS  &C758
C717 38           8       SEC
C718 ED 20 03     . .     SBC  &0320
C71B 30 3B        0;      BMI  &C758
C71D A8           .       TAY
C71E 20 88 CA      ..     JSR  &CA88
C721 A9 08        ..      LDA  #&08
C723 20 9D C5      ..     JSR  &C59D
C726 A2 20        .       LDX  #&20
C728 A0 08        ..      LDY  #&08
C72A 20 8A D4      ..     JSR  &D48A
C72D 20 E8 CE      ..     JSR  &CEE8
C730 B0 47        .G      BCS  &C779
C732 4C 02 CA     L..     JMP  &CA02
```

OSWORD 9 Read pixel value.

```
C735 A0 03        ..      LDY  #&03
C737 B1 F0        ..      LDA  (&F0),Y
C739 99 28 03     .(.     STA  &0328,Y
C73C 88           .       DEY
C73D 10 F8        ..      BPL  &C737
C73F A9 28        .(      LDA  #&28
C741 20 39 D8      9.     JSR  &D839
C744 A0 04        ..      LDY  #&04
C746 D0 08        ..      BNE  &C750
```

OSWORD &B Read palette.

```
C748 2D 60 03     -`.     AND  &0360
C74B AA           .       TAX
C74C BD 6F 03     .o.     LDA  &036F,X
C74F C8           .       INY
C750 91 F0        ..      STA  (&F0),Y
C752 A9 00        ..      LDA  #&00
C754 C0 04        ..      CPY  #&04
C756 D0 F7        ..      BNE  &C74F
C758 60           `       RTS
```

VDU 12

```
C759 20 88 C5      ..     JSR  &C588
C75C D0 5F        ._      BNE  &C7BD
C75E A5 D0        ..      LDA  &D0
C760 29 08        ).      AND  #&08
C762 D0 03        ..      BNE  &C767
C764 4C C1 CB     L..     JMP  &CBC1
C767 AE 0B 03     ...     LDX  &030B
C76A 8E 19 03     ...     STX  &0319
C76D 20 AC CE      ..     JSR  &CEAC
```

```
C770 AE 19 03     ...     LDX  &0319
C773 EC 09 03     ...     CPX  &0309
C776 E8           .       INX
C777 90 F1        ..      BCC  &C76A
C779 20 88 C5      ..     JSR  &C588
C77C F0 03        ..      BEQ  &C781
C77E 4C A6 CF     L..     JMP  &CFA6
C781 8D 23 03     .#.     STA  &0323
C784 8D 22 03     ."."    STA  &0322
C787 20 88 C5      ..     JSR  &C588
C78A D0 CC        ..      BNE  &C758
C78C 20 A8 C7      ..     JSR  &C7A8
C78F 18           .       CLC
C790 AD 22 03     ."."    LDA  &0322
C793 6D 08 03     m..     ADC  &0308
C796 8D 18 03     ...     STA  &0318
C799 AD 23 03     .#.     LDA  &0323
C79C 18           .       CLC
C79D 6D 0B 03     m..     ADC  &030B
C7A0 8D 19 03     ...     STA  &0319
C7A3 20 E8 CE      ..     JSR  &CEE8
C7A6 90 8A        ..      BCC  &C732
C7A8 A2 18        ..      LDX  #&18
C7AA A0 28        .(      LDY  #&28
C7AC 4C DE CD     L..     JMP  &CDDE
C7AF 20 88 C5      ..     JSR  &C588
C7B2 F0 03        ..      BEQ  &C7B7
C7B4 4C AD CF     L..     JMP  &CFAD
C7B7 20 6E CE      n.     JSR  &CE6E
C7BA 4C AF C6     L..     JMP  &C6AF
C7BD 20 A6 CF      ..     JSR  &CFA6
C7C0 AD 61 03     .a.     LDA  &0361
C7C3 F0 33        .3      BEQ  &C7F8
C7C5 AE 5A 03     .Z.     LDX  &035A
C7C8 AC 5C 03     .\.     LDY  &035C
C7CB 20 B3 D0      ..     JSR  &D0B3
C7CE A2 00        ..      LDX  #&00
C7D0 A0 28        .(      LDY  #&28
C7D2 20 7C D4      |.     JSR  &D47C
C7D5 38           8       SEC
C7D6 AD 06 03     ...     LDA  &0306
C7D9 ED 02 03     ...     SBC  &0302
C7DC A8           .       TAY
C7DD C8           .       INY
C7DE 8C 30 03     .0.     STY  &0330
C7E1 A2 2C        .,      LDX  #&2C
C7E3 A0 28        .(      LDY  #&28
C7E5 20 A6 D6      ..     JSR  &D6A6
C7E8 AD 2E 03     ...     LDA  &032E
C7EB D0 03        ..      BNE  &C7F0
C7ED CE 2F 03     ./.     DEC  &032F
C7F0 CE 2E 03     ...     DEC  &032E
C7F3 CE 30 03     .0.     DEC  &0330
C7F6 D0 E9        ..      BNE  &C7E1
C7F8 60           `       RTS
C7F9 A0 00        ..      LDY  #&00
C7FB F0 02        ..      BEQ  &C7FF
C7FD A0 02        ..      LDY  #&02
C7FF AD 23 03     .#.     LDA  &0323
C802 10 01        ..      BPL  &C805
C804 C8           .       INY
C805 2D 60 03     -`.     AND  &0360
```

VDU 30

VDU 13

VDU 16

```
C808 85 DA        ..      STA  &DA
C80A AD 60 03     .`.     LDA  &0360
C80D F0 1C        ..      BEQ  &C82B
C80F 29 07        ).      AND  #&07
C811 18           .       CLC
C812 65 DA        e.      ADC  &DA
C814 AA           .       TAX
C815 BD 23 C4     .#.     LDA  &C423,X
C818 99 57 03     .W.     STA  &0357,Y
C81B C0 02        ..      CPY  #&02
C81D B0 0D        ..      BCS  &C82C
C81F AD 57 03     .W.     LDA  &0357
C822 49 FF        I.      EOR  #&FF
C824 85 D3        ..      STA  &D3
C826 4D 58 03     MX.     EOR  &0358
C829 85 D2        ..      STA  &D2
C82B 60           `       RTS
C82C AD 22 03     ."."    LDA  &0322
C82F 99 59 03     .Y.     STA  &0359,Y
C832 60           `       RTS
C833 A9 20        .       LDA  #&20   } For Mode 7, background text col = &20.
C835 8D 58 03     .X.     STA  &0358  }
C838 60           `       RTS
C839 A2 05        ..      LDX  #&05   } &357 - &35C = 0.
C83B A9 00        ..      LDA  #&00   }
C83D 9D 57 03     .W.     STA  &0357,X}
C840 CA           .       DEX
C841 10 FA        ..      BPL  &C83D
C843 AE 60 03     .`.     LDX  &0360  No of colours (-1) in current mode.
C846 F0 EB        ..      BEQ  &C833  If Mode 7 (0) then branch
C848 A9 FF        ..      LDA  #&FF
C84A E0 0F        ..      CPX  #&0F   } 15 cols? branch if not Mode 2.
C84C D0 02        ..      BNE  &C850  }
C84E A9 3F        .?      LDA  #&3F   A = &3F
C850 8D 57 03     .W.     STA  &0357  } Foreground text colour.  (&3F = Mode 2
C853 8D 59 03     .Y.     STA  &0359  } Foreground graphics colour.  &7F <> Mode 2)
C856 49 FF        I.      EOR  #&FF   (Not) all bits.
C858 85 D2        ..      STA  &D2    } text colour bytes to be OR'ed & EOR'ed Respectively
C85A 85 D3        ..      STA  &D3    }
C85C 8E 1F 03     ...     STX  &031F  No of colours (-1) put in VDU queue.?
C85F E0 03        ..      CPX  #&03   } 4 colour Mode?, branch if true.
C861 F0 11        ..      BEQ  &C874  }
C863 90 20        .       BCC  &C885  }
C865 8E 20 03     . .     STX  &0320  16 cols. (2).
C868 20 92 C8      ..     JSR  &C892
C86B CE 20 03     . .     DEC  &0320
C86E CE 1F 03     ...     DEC  &031F
C871 10 F5        ..      BPL  &C868
C873 60           `       RTS
C874 A2 07        ..      LDX  #&07   For 4 cols. (1,5)
C876 8E 20 03     . .     STX  &0320
C879 20 92 C8      ..     JSR  &C892
C87C 4E 20 03     N .     LSR  &0320
C87F CE 1F 03     ...     DEC  &031F
C882 10 F5        ..      BPL  &C879
C884 60           `       RTS
C885 A2 07        ..      LDX  #&07   For 2 cols. (0,3,6)
C887 20 8F C8      ..     JSR  &C88F
C88A A2 00        ..      LDX  #&00
C88C 8E 1F 03     ...     STX  &031F
C88F 8E 20 03     . .     STX  &0320
C892 08           .       PHP
```

&VDU 20

```
C893 78          x      SEI
C894 AD 1F 03    ...    LDA  &031F
C897 2D 60 03    -`.    AND  &0360
C89A AA          .      TAX
C89B AD 20 03    . .    LDA  &0320
C89E 29 0F       ).     AND  #&0F
C8A0 9D 6F 03    .o.    STA  &036F,X
C8A3 A8          .      TAY
C8A4 AD 60 03    .`.    LDA  &0360
C8A7 85 FA       ..     STA  &FA
C8A9 C9 03       ..     CMP  #&03
C8AB 08          .      PHP
C8AC 8A          .      TXA
C8AD 6A          j      ROR  A
C8AE 66 FA       f.     ROR  &FA
C8B0 B0 FB       ..     BCS  &C8AD
C8B2 06 FA       ..     ASL  &FA
C8B4 98          .      TYA
C8B5 05 FA       ..     ORA  &FA
C8B7 AA          .      TAX
C8B8 A0 00       ..     LDY  #&00
C8BA 28          (      PLP
C8BB 08          .      PHP
C8BC D0 0E       ..     BNE  &C8CC
C8BE 29 60       )`     AND  #&60
C8C0 F0 09       ..     BEQ  &C8CB
C8C2 C9 60       .`     CMP  #&60
C8C4 F0 05       ..     BEQ  &C8CB
C8C6 8A          .      TXA
C8C7 49 60       I`     EOR  #&60
C8C9 D0 01       ..     BNE  &C8CC
C8CB 8A          .      TXA
C8CC 20 11 EA     ..    JSR  &EA11
C8CF 98          .      TYA
C8D0 38          8      SEC
C8D1 6D 60 03    m`.    ADC  &0360
C8D4 A8          .      TAY
C8D5 8A          .      TXA
C8D6 69 10       i.     ADC  #&10
C8D8 AA          .      TAX
C8D9 C0 10       ..     CPY  #&10
C8DB 90 DD       ..     BCC  &C8BA
C8DD 28          (      PLP
C8DE 28          (      PLP
C8DF 60          `      RTS
```

OSWORD &C Write palette.

```
C8E0 08          .      PHP
C8E1 2D 60 03    -`.    AND  &0360
C8E4 AA          .      TAX
C8E5 C8          .      INY
C8E6 B1 F0       ..     LDA  (&F0),Y
C8E8 4C 9E C8    L..    JMP  &C89E
C8EB AD 23 03    .#.    LDA  &0323
C8EE 4C 33 CB    L3.    JMP  &CB33
C8F1 AD 1B 03    ...    LDA  &031B
C8F4 C9 20       .      CMP  #&20
C8F6 90 47       .G     BCC  &C93F
C8F8 48          H      PHA
C8F9 4A          J      LSR  A
C8FA 4A          J      LSR  A
C8FB 4A          J      LSR  A
C8FC 4A          J      LSR  A
C8FD 4A          J      LSR  A
```

```
C8FE AA          .      TAX
C8FF BD 0D C4    ...    LDA  &C40D,X
C902 2C 67 03    ,g.    BIT  &0367
C905 D0 20       .      BNE  &C927
C907 0D 67 03    .g.    ORA  &0367
C90A 8D 67 03    .g.    STA  &0367
C90D 8A          .      TXA
C90E 29 03       ).     AND  #&03
C910 18          .      CLC
C911 69 BF       i.     ADC  #&BF
C913 85 DF       ..     STA  &DF
C915 BD 67 03    .g.    LDA  &0367,X
C918 85 DD       ..     STA  &DD
C91A A0 00       ..     LDY  #&00
C91C 84 DC       ..     STY  &DC
C91E 84 DE       ..     STY  &DE
C920 B1 DE       ..     LDA  (&DE),Y
C922 91 DC       ..     STA  (&DC),Y
C924 88          .      DEY
C925 D0 F9       ..     BNE  &C920
C927 68          h      PLA
C928 20 3E D0     >.    JSR  &D03E
C92B A0 07       ..     LDY  #&07
C92D B9 1C 03    ...    LDA  &031C,Y
C930 91 DE       ..     STA  (&DE),Y
C932 88          .      DEY
C933 10 F8       ..     BPL  &C92D
C935 60          `      RTS
C936 68          h      PLA
C937 60          `      RTS
C938 AD 1F 03    ...    LDA  &031F
C93B 18          .      CLC
C93C 6C 26 02    l&.    JMP  (&0226)
C93F C9 01       ..     CMP  #&01
C941 90 15       ..     BCC  &C958
C943 D0 F7       ..     BNE  &C93C
C945 20 88 C5     ..    JSR  &C588
C948 D0 ED       ..     BNE  &C937
C94A A9 20       .      LDA  #&20
C94C AC 1C 03    ...    LDY  &031C
C94F F0 03       ..     BEQ  &C954
C951 AD 5F 03    ._.    LDA  &035F
C954 A0 0A       ..     LDY  #&0A
C956 D0 2D       .-     BNE  &C985
C958 AD 1D 03    ...    LDA  &031D
C95B AC 1C 03    ...    LDY  &031C
C95E C0 07       ..     CPY  #&07
C960 90 23       .#     BCC  &C985
C962 D0 03       ..     BNE  &C967
C964 6D 90 02    m..    ADC  &0290
C967 C0 08       ..     CPY  #&08
C969 D0 07       ..     BNE  &C972
C96B 09 00       ..     ORA  #&00
C96D 30 03       0.     BMI  &C972
C96F 4D 91 02    M..    EOR  &0291
C972 C0 0A       ..     CPY  #&0A
C974 D0 0F       ..     BNE  &C985
C976 8D 5F 03    ._.    STA  &035F
C979 A8          .      TAY
C97A A5 D0       ..     LDA  &D0
C97C 29 20       )      AND  #&20
C97E 08          .      PHP
```

The PLOT number last issued (ie in the VDU queue.)

C=0 VDU extension vector

Last OSBYTE answer font setting

} TAKE always (write to register &A)

```
C97F 98          .      TYA
C980 A0 0A       ..     LDY  #&0A
C982 28          (      PLP
C983 D0 06       ..     BNE  &C98B
C985 8C 00 FE    ...    STY  &FE00
C988 8D 01 FE    ...    STA  &FE01
C98B 60          `      RTS
C98C AE 61 03    .a.    LDX  &0361
C98F F0 A7       ..     BEQ  &C938
C991 4C 60 D0    L`.    JMP  &D060
C994 AE 50 03    .P.    LDX  &0350
C997 AD 51 03    .Q.    LDA  &0351
C99A 20 F8 CC     ..    JSR  &CCF8
C99D B0 14       ..     BCS  &C9B3
C99F 6D 54 03    mT.    ADC  &0354
C9A2 90 0F       ..     BCC  &C9B3
C9A4 AE 50 03    .P.    LDX  &0350
C9A7 AD 51 03    .Q.    LDA  &0351
C9AA 20 D4 CA     ..    JSR  &CAD4
C9AD 10 04       ..     BPL  &C9B3
C9AF 38          8      SEC
C9B0 ED 54 03    .T.    SBC  &0354
C9B3 8D 51 03    .Q.    STA  &0351
C9B6 8E 50 03    .P.    STX  &0350
C9B9 A0 0C       ..     LDY  #&0C
C9BB D0 51       .Q     BNE  &CA0E
C9BD A9 00       ..     LDA  #&00
C9BF A2 2C       .,     LDX  #&2C
C9C1 9D 00 03    ...    STA  &0300,X
C9C4 CA          .      DEX
C9C5 10 FA       ..     BPL  &C9C1
C9C7 AE 55 03    .U.    LDX  &0355
C9CA BC EF C3    ...    LDY  &C3EF,X
C9CD 8C 0A 03    ...    STY  &030A
C9D0 20 88 CA     ..    JSR  &CA88
C9D3 BC E7 C3    ...    LDY  &C3E7,X
C9D6 8C 09 03    ...    STY  &0309
C9D9 A0 03       ..     LDY  #&03
C9DB 8C 23 03    .#.    STY  &0323
C9DE C8          .      INY
C9DF 8C 21 03    .!.    STY  &0321
C9E2 CE 22 03    ."     DEC  &0322
C9E5 CE 20 03    . .    DEC  &0320
C9E8 20 39 CA     9.    JSR  &CA39
C9EB A9 F7       ..     LDA  #&F7
C9ED 20 A8 C5     ..    JSR  &C5A8
C9F0 AE 50 03    .P.    LDX  &0350
C9F3 AD 51 03    .Q.    LDA  &0351
C9F6 8E 4A 03    .J.    STX  &034A
C9F9 8D 4B 03    .K.    STA  &034B
C9FC 10 04       ..     BPL  &CA02
C9FE 38          8      SEC
C9FF ED 54 03    .T.    SBC  &0354
CA02 86 D8       ..     STX  &D8
CA04 85 D9       ..     STA  &D9
CA06 AE 4A 03    .J.    LDX  &034A
CA09 AD 4B 03    .K.    LDA  &034B
CA0C A0 0E       ..     LDY  #&0E
CA0E 48          H      PHA
CA0F AD 55 03    .U.    LDA  &0355
CA12 C9 07       ..     CMP  #&07
CA14 68          h      PLA
```

PLOT K,x,y. entry.

No of pixels per byte (-1) — this goes to the VDU extension vector at &226 — take branch only if in text

Start of PLOT decode routine.

VDU 26

restore text cursor low & high for 6845

if high is +ve branch.

high byte of size of screen memory (eg Ø,1 = 2 = 8

copy text cursor low & high for 6845

retrieve cursor position for 6845 ?

Y = &E

save A - high of cursor position.

Screen mode

Mode 7 ?

retrieve A.

```
CA15 B0 10        ..    BCS  &CA27
CA17 86 DA        ..    STX  &DA
CA19 4A           J     LSR  A
CA1A 66 DA        f.    ROR  &DA
CA1C 4A           J     LSR  A
CA1D 66 DA        f.    ROR  &DA
CA1F 4A           J     LSR  A
CA20 66 DA        f.    ROR  &DA
CA22 A6 DA        ..    LDX  &DA
CA24 4C 2B CA     L+.   JMP  &CA2B
CA27 E9 74        .t    SBC  #&74
CA29 49 20        I     EOR  #&20
CA2B 8C 00 FE     ...   STY  &FE00
CA2E 8D 01 FE     ...   STA  &FE01
CA31 C8           .     INY
CA32 8C 00 FE     ...   STY  &FE00
CA35 8E 01 FE     ...   STX  &FE01
CA38 60           `     RTS
CA39 20 81 CA      ..   JSR  &CA81
CA3C A2 1C        ..    LDX  #&1C
CA3E A0 2C        .,    LDY  #&2C
CA40 20 11 D4      ..   JSR  &D411
CA43 0D 2D 03     .-.   ORA  &032D
CA46 30 39        09    BMI  &CA81
CA48 A2 20        .     LDX  #&20
CA4A 20 49 D1      I.   JSR  &D149
CA4D A2 1C        ..    LDX  #&1C
CA4F 20 49 D1      I.   JSR  &D149
CA52 AD 1F 03     ...   LDA  &031F
CA55 0D 1D 03     ...   ORA  &031D
CA58 30 27        0'    BMI  &CA81
CA5A AD 23 03     .#.   LDA  &0323
CA5D D0 22        ."    BNE  &CA81
CA5F AE 55 03     .U.   LDX  &0355
CA62 AD 21 03     .!.   LDA  &0321
CA65 85 DA        ..    STA  &DA
CA67 AD 20 03     . .   LDA  &0320
CA6A 46 DA        F.    LSR  &DA
CA6C 6A           j     ROR  A
CA6D 46 DA        F.    LSR  &DA
CA6F D0 10        ..    BNE  &CA81
CA71 6A           j     ROR  A
CA72 4A           J     LSR  A
CA73 DD EF C3     ...   CMP  &C3EF,X
CA76 F0 02        ..    BEQ  &CA7A
CA78 10 07        ..    BPL  &CA81
CA7A A0 00        ..    LDY  #&00
CA7C A2 1C        ..    LDX  #&1C
CA7E 20 7C D4      |.   JSR  &D47C
CA81 A2 10        ..    LDX  #&10
CA83 A0 28        .(    LDY  #&28
CA85 4C E6 CD     L..   JMP  &CDE6
CA88 C8           .     INY
CA89 98           .     TYA
CA8A A0 00        ..    LDY  #&00
CA8C 8C 4D 03     .M.   STY  &034D
CA8F 8D 4C 03     .L.   STA  &034C
CA92 AD 4F 03     .O.   LDA  &034F
CA95 4A           J     LSR  A
CA96 F0 09        ..    BEQ  &CAA1
CA98 0E 4C 03     .L.   ASL  &034C
CA9B 2E 4D 03     .M.   ROL  &034D
```

branch if Mode 7 or >.
text cursor low byte ( temp workspace
/2
/2
/2
/2
/2
/2
Fudge for Mode 7.
Y= neg 14 ↓15, X= high } A= low } text cursor position
write to 6845
exit

```
CA9E 4A          J       LSR  A
CA9F 90 F7       ..      BCC  &CA98
CAA1 60          `       RTS
CAA2 A2 20       .       LDX  #&20
CAA4 A0 0C       ..      LDY  #&0C
CAA6 20 8A D4     ..     JSR  &D48A
CAA9 4C B8 D1    L..     JMP  &D1B8
```

VDU 127 (delete).

```
CAAC 20 C5 C5     ..     JSR  &C5C5
CAAF 20 88 C5     ..     JSR  &C588
CAB2 D0 13       ..      BNE  &CAC7
CAB4 AE 60 03    .`.     LDX  &0360
CAB7 F0 09       ..      BEQ  &CAC2
CAB9 85 DE       ..      STA  &DE
CABB A9 C0       ..      LDA  #&C0
CABD 85 DF       ..      STA  &DF
CABF 4C BF CF    L..     JMP  &CFBF
CAC2 A9 20       .       LDA  #&20
CAC4 4C DC CF    L..     JMP  &CFDC
CAC7 A9 7F       ..      LDA  #&7F
CAC9 20 3E D0     >.     JSR  &D03E
CACC AE 5A 03    .Z.     LDX  &035A
CACF A0 00       ..      LDY  #&00
CAD1 4C 63 CF    Lc.     JMP  &CF63
CAD4 48          H       PHA
CAD5 8A          .       TXA
CAD6 18          .       CLC
CAD7 6D 52 03    mR.     ADC  &0352
CADA AA          .       TAX
CADB 68          h       PLA
CADC 6D 53 03    mS.     ADC  &0353
CADF 60          `       RTS
CAE0 20 14 CB     ..     JSR  &CB14
CAE3 20 D9 E9     ..     JSR  &E9D9
CAE6 90 02       ..      BCC  &CAEA
CAE8 30 F6       0.      BMI  &CAE0
CAEA A5 D0       ..      LDA  &D0
CAEC 49 04       I.      EOR  #&04
CAEE 29 46       )F      AND  #&46
CAF0 D0 2A       .*      BNE  &CB1C
CAF2 AD 69 02    .i.     LDA  &0269
CAF5 30 22       0"      BMI  &CB19
CAF7 AD 19 03    ...     LDA  &0319
CAFA CD 09 03    ...     CMP  &0309
CAFD 90 1A       ..      BCC  &CB19
CAFF 4A          J       LSR  A
CB00 4A          J       LSR  A
CB01 38          8       SEC
CB02 6D 69 02    mi.     ADC  &0269
CB05 6D 0B 03    m..     ADC  &030B
CB08 CD 09 03    ...     CMP  &0309
CB0B 90 0C       ..      BCC  &CB19
CB0D 18          .       CLC
CB0E 20 D9 E9     ..     JSR  &E9D9
CB11 38          8       SEC
CB12 10 FA       ..      BPL  &CB0E
CB14 A9 FF       ..      LDA  #&FF
CB16 8D 69 02    .i.     STA  &0269
CB19 EE 69 02    .i.     INC  &0269
CB1C 60          `       RTS
```

Save mode number. Change Mode

```
CB1D 48          H       PHA
CB1E A2 7F       ..      LDX  #&7F
CB20 A9 00       ..      LDA  #&00
```

Acc = Mode Bits 0-2.

X=&7F A=Ø.

```
CB22 85 D0      ..      STA  &D0        VDU Status
CB24 9D FF 02   ...     STA  &02FF,X    } &3ØØ-&37E (incl) = Ø
CB27 CA         .       DEX             }
CB28 D0 FA      ..      BNE  &CB24      }
CB2A 20 07 CD     ..    JSR  &CD07      explode sgt character RAM allocation.
CB2D 68         h       PLA             retrieve mode number.
CB2E A2 7F      ..      LDX  #&7F       } character to be used for output cursor (Mode 7
CB30 8E 66 03   .f.     STX  &0366      }
CB33 2C 8E 02   ,..     BIT  &028E      } available ram, branch if 32K.
CB36 30 02      0.      BMI  &CB3A      }
CB38 09 04      ..      ORA  #&04       16K adjustment to mode number.
CB3A 29 07      ).      AND  #&07       get bits Ø-2
CB3C AA         .       TAX             X= Mode number
CB3D 8E 55 03   .U.     STX  &0355      Store Mode number in &355
CB40 BD 14 C4   ...     LDA  &C414,X    } read & save No of colours(-1) for Mode.
CB43 8D 60 03   .`.     STA  &0360      }
CB46 BD FF C3   ...     LDA  &C3FF,X    }  "   "   "  No of bytes taken per character.
CB49 8D 4F 03   .O.     STA  &034F      }
CB4C BD 3A C4   .:.     LDA  &C43A,X    }  "   "   "  No of pixels per byte (-1) for mode
CB4F 8D 61 03   .a.     STA  &0361      }
CB52 D0 02      ..      BNE  &CB56      branch if Mode Ø-5 incl
CB54 A9 07      ..      LDA  #&07       6&7 only.
CB56 0A         .       ASL  A          *2
CB57 A8         .       TAY             Y=Acc
CB58 B9 06 C4   ...     LDA  &C406,Y
CB5B 8D 63 03   .c.     STA  &0363
CB5E 0A         .       ASL  A          *2
CB5F 10 FD      ..      BPL  &CB5E
CB61 8D 62 03   .b.     STA  &0362
CB64 BC 40 C4   .@.     LDY  &C440,X    } Memory Map type   Y=0=20K Y=1=16K Y=4=16
CB67 8C 56 03   .V.     STY  &0356      }                   Y=2=10K Y=3=8K
CB6A B9 4F C4   .O.     LDA  &C44F,Y    } various VDU control & put in output register B
CB6D 20 F8 E9     ..    JSR  &E9F8      }   numbers.                    System VIA.
CB70 B9 4B C4   .K.     LDA  &C44B,Y    }  as above                     (screen rap around)
CB73 20 F8 E9     ..    JSR  &E9F8      }                               size for
CB76 B9 59 C4   .Y.     LDA  &C459,Y    } high byte of screen size.     6845
CB79 8D 54 03   .T.     STA  &0354      }                               P419
CB7C B9 5E C4   .^.     LDA  &C45E,Y    } high byte of screen start address.  AM
CB7F 8D 4E 03   .N.     STA  &034E      }
CB82 98         .       TYA             A= Ø-4 (incl).
CB83 69 02      i.      ADC  #&02       Ac= 2-6 (incl)   ? carry.
CB85 49 07      I.      EOR  #&07       EOR # %111
CB87 4A         J       LSR  A          /2
CB88 AA         .       TAX             X= Acc
CB89 BD 66 C4   .f.     LDA  &C466,X    } low byte of pointer to start of multiplication table.
CB8C 85 E0      ..      STA  &E0        }
CB8E A9 C3      ..      LDA  #&C3       } high byte of pointer to start of Multiplication tabl
CB90 85 E1      ..      STA  &E1        }
CB92 BD 63 C4   .c.     LDA  &C463,X    }
CB95 8D 52 03   .R.     STA  &0352      } LSB of size of one line   &280-0123  &28-7
CB98 8E 53 03   .S.     STX  &0353      }                           &140-3456
CB9B A9 43      .C      LDA  #&43       } change VDU status to &43.
CB9D 20 A8 C5     ..    JSR  &C5A8      }
CBA0 AE 55 03   .U.     LDX  &0355      screen mode.
CBA3 BD F7 C3   ...     LDA  &C3F7,X    } write to Video ULA the correct value for
CBA6 20 00 EA     ..    JSR  &EA00      }   given mode.
CBA9 08         .       PHP
CBAA 78         x       SEI             (Y in range Ø-4) check!
CBAB BE 69 C4   .i.     LDX  &C469,Y
CBAE A0 0B      ..      LDY  #&0B
CBB0 BD 6E C4   .n.     LDA  &C46E,X    6845 values for Mode (Ø-7)
CBB3 20 5E C9     ^.    JSR  &C95E      write to 6845.
```

```
CBB6 CA          .      DEX
CBB7 88          .      DEY
CBB8 10 F6       ..     BPL  &CBB0
CBBA 28          (      PLP
CBBB 20 39 C8     9.    JSR  &C839
CBBE 20 BD C9     ..    JSR  &C9BD
CBC1 A2 00       ..     LDX  #&00
CBC3 AD 4E 03    .N.    LDA  &034E
CBC6 8E 50 03    .P.    STX  &0350
CBC9 8D 51 03    .Q.    STA  &0351
CBCC 20 F6 C9     ..    JSR  &C9F6
CBCF A0 0C       ..     LDY  #&0C
CBD1 20 2B CA     +.    JSR  &CA2B
CBD4 AD 58 03    .X.    LDA  &0358
CBD7 AE 56 03    .V.    LDX  &0356
CBDA BC 54 C4    .T.    LDY  &C454,X
CBDD 8C 5D 03    .].    STY  &035D
CBE0 A0 CC       ..     LDY  #&CC
CBE2 8C 5E 03    .^.    STY  &035E
CBE5 A2 00       ..     LDX  #&00
CBE7 8E 69 02    .i.    STX  &0269
CBEA 8E 18 03    ...    STX  &0318
CBED 8E 19 03    ...    STX  &0319
CBF0 6C 5D 03    l].    JMP  (&035D)
CBF3 20 3E D0     >.    JSR  &D03E
CBF6 A0 00       ..     LDY  #&00
CBF8 B1 DE       ..     LDA  (&DE),Y
CBFA C8          .      INY
CBFB 91 F0       ..     STA  (&F0),Y
CBFD C0 08       ..     CPY  #&08
CBFF D0 F7       ..     BNE  &CBF8
CC01 60          `      RTS
CC02 9D 00 30    ..0    STA  &3000,X
CC05 9D 00 31    ..1    STA  &3100,X
CC08 9D 00 32    ..2    STA  &3200,X
CC0B 9D 00 33    ..3    STA  &3300,X
CC0E 9D 00 34    ..4    STA  &3400,X
CC11 9D 00 35    ..5    STA  &3500,X
CC14 9D 00 36    ..6    STA  &3600,X
CC17 9D 00 37    ..7    STA  &3700,X
CC1A 9D 00 38    ..8    STA  &3800,X
CC1D 9D 00 39    ..9    STA  &3900,X
CC20 9D 00 3A    ..:    STA  &3A00,X
CC23 9D 00 3B    ..;    STA  &3B00,X
CC26 9D 00 3C    ..<    STA  &3C00,X
CC29 9D 00 3D    ..=    STA  &3D00,X
CC2C 9D 00 3E    ..>    STA  &3E00,X
CC2F 9D 00 3F    ..?    STA  &3F00,X
CC32 9D 00 40    ..@    STA  &4000,X
CC35 9D 00 41    ..A    STA  &4100,X
CC38 9D 00 42    ..B    STA  &4200,X
CC3B 9D 00 43    ..C    STA  &4300,X
CC3E 9D 00 44    ..D    STA  &4400,X
CC41 9D 00 45    ..E    STA  &4500,X
CC44 9D 00 46    ..F    STA  &4600,X
CC47 9D 00 47    ..G    STA  &4700,X
CC4A 9D 00 48    ..H    STA  &4800,X
CC4D 9D 00 49    ..I    STA  &4900,X
CC50 9D 00 4A    ..J    STA  &4A00,X
CC53 9D 00 4B    ..K    STA  &4B00,X
CC56 9D 00 4C    ..L    STA  &4C00,X
CC59 9D 00 4D    ..M    STA  &4D00,X
```

repeat until All regs done (0-11) incl.

! ½ done.
! to be done.
X=0
high byte of Screen Start address.
low of Screen Start. = 0
copy of high byte for Screen Start.
! to be done.
Y= register (6845) Start register.
write top left Screen data to 6845.
background text colour.
Memory Map type.
Misc VDU codes.
low byte of JMP Vector For CLS. } see
high " " " " " " . } &CC02+.

No of lines since last page halt.
current text cursor X & Y positions, respectively
Jump to CLS Start.

OSWORD &A Read character Definition

012

CLS Start Address'

| | |
|---|---|
| &CC02 | Modes 0, 1, 2. |
| &CC32 | Mode 3. |
| &CC7A | Modes 4 & 5. |

Clear Screen

| | |
|---|---|
| &CC92 | Mode 6. |
| &CCE6 | Mode 7. |

3 (CLS)

```
CC5C 9D 00 4E     ..N     STA  &4E00,X
CC5F 9D 00 4F     ..O     STA  &4F00,X
CC62 9D 00 50     ..P     STA  &5000,X
CC65 9D 00 51     ..Q     STA  &5100,X
CC68 9D 00 52     ..R     STA  &5200,X
CC6B 9D 00 53     ..S     STA  &5300,X
CC6E 9D 00 54     ..T     STA  &5400,X
CC71 9D 00 55     ..U     STA  &5500,X
CC74 9D 00 56     ..V     STA  &5600,X
CC77 9D 00 57     ..W     STA  &5700,X
CC7A 9D 00 58     ..X     STA  &5800,X
CC7D 9D 00 59     ..Y     STA  &5900,X
CC80 9D 00 5A     ..Z     STA  &5A00,X
CC83 9D 00 5B     ..[     STA  &5B00,X
CC86 9D 00 5C     ..\     STA  &5C00,X
CC89 9D 00 5D     ..]     STA  &5D00,X
CC8C 9D 00 5E     ..^     STA  &5E00,X
CC8F 9D 00 5F     .._     STA  &5F00,X
CC92 9D 00 60     ..`     STA  &6000,X
CC95 9D 00 61     ..a     STA  &6100,X
CC98 9D 00 62     ..b     STA  &6200,X
CC9B 9D 00 63     ..c     STA  &6300,X
CC9E 9D 00 64     ..d     STA  &6400,X
CCA1 9D 00 65     ..e     STA  &6500,X
CCA4 9D 00 66     ..f     STA  &6600,X
CCA7 9D 00 67     ..g     STA  &6700,X
CCAA 9D 00 68     ..h     STA  &6800,X
CCAD 9D 00 69     ..i     STA  &6900,X
CCB0 9D 00 6A     ..j     STA  &6A00,X
CCB3 9D 00 6B     ..k     STA  &6B00,X
CCB6 9D 00 6C     ..l     STA  &6C00,X
CCB9 9D 00 6D     ..m     STA  &6D00,X
CCBC 9D 00 6E     ..n     STA  &6E00,X
CCBF 9D 00 6F     ..o     STA  &6F00,X
CCC2 9D 00 70     ..p     STA  &7000,X
CCC5 9D 00 71     ..q     STA  &7100,X
CCC8 9D 00 72     ..r     STA  &7200,X
CCCB 9D 00 73     ..s     STA  &7300,X
CCCE 9D 00 74     ..t     STA  &7400,X
CCD1 9D 00 75     ..u     STA  &7500,X
CCD4 9D 00 76     ..v     STA  &7600,X
CCD7 9D 00 77     ..w     STA  &7700,X
CCDA 9D 00 78     ..x     STA  &7800,X
CCDD 9D 00 79     ..y     STA  &7900,X
CCE0 9D 00 7A     ..z     STA  &7A00,X
CCE3 9D 00 7B     ..{     STA  &7B00,X
CCE6 9D 00 7C     ..|     STA  &7C00,X
CCE9 9D 00 7D     ..}     STA  &7D00,X
CCEC 9D 00 7E     ..~     STA  &7E00,X
CCEF 9D 00 7F     ...     STA  &7F00,X
CCF2 E8           .       INX
CCF3 F0 70        .p      BEQ  &CD65   Exit Mode change
CCF5 6C 5D 03     l].     JMP  (&035D) Next offset, From Start
CCF8 48           H       PHA
CCF9 8A           .       TXA
CCFA 38           8       SEC
CCFB ED 52 03     .R.     SBC  &0352
CCFE AA           .       TAX
CCFF 68           h       PLA
CD00 ED 53 03     .S.     SBC  &0353
CD03 CD 4E 03     .N.     CMP  &034E
CD06 60           `       RTS
```

485

(CLS)

continued.

6

7

```
CD07 A9 0F       ..     LDA  #&0F      Font flag=&F   *FX 20   Explode soft character RAM
CD09 8D 67 03    .g.    STA  &0367
CD0C A9 0C       ..     LDA  #&0C
CD0E A0 06       ..     LDY  #&06      &36E - &368 = &C (ie page &C) allocation
CD10 99 68 03    .h.    STA  &0368,Y
CD13 88          .      DEY
CD14 10 FA       ..     BPL  &CD10
CD16 E0 07       ..     CPX  #&07      Make X=6 if greater than 6
CD18 90 02       ..     BCC  &CD1C
CD1A A2 06       ..     LDX  #&06
CD1C 8E 46 02    .F.    STX  &0246     Save Font explosion state
CD1F AD 43 02    .C.    LDA  &0243     primary OSHWM
CD22 A2 00       ..     LDX  #&00      X=0
CD24 EC 46 02    .F.    CPX  &0246     branch if last page being reserved has been pro
CD27 B0 0B       ..     BCS  &CD34
CD29 BC BA C4    ...    LDY  &C4BA,X   Y=
CD2C 99 68 03    .h.    STA  &0368,Y   save page
CD2F 69 01       i.     ADC  #&01      +1
CD31 E8          .      INX
CD32 D0 F0       ..     BNE  &CD24
CD34 8D 44 02    .D.    STA  &0244     OSHWM + sgt beg data room.
CD37 A8          .      TAY
CD38 F0 CC       ..     BEQ  &CD06     exit if OSHWM+sgt beg=0 !?
CD3A A2 11       ..     LDX  #&11
CD3C 4C 68 F1    Lh.    JMP  &F168     Issue service call &11 - Font explosion/impl
                                                                  warning
CD3F A9 02       ..     LDA  #&02
CD41 24 D0       $.     BIT  &D0
CD43 D0 02       ..     BNE  &CD47
CD45 50 32       P2     BVC  &CD79
CD47 AD 09 03    ...    LDA  &0309
CD4A 90 03       ..     BCC  &CD4F
CD4C AD 0B 03    ...    LDA  &030B
CD4F 70 08       p.     BVS  &CD59
CD51 8D 19 03    ...    STA  &0319
CD54 68          h      PLA
CD55 68          h      PLA
CD56 4C AF C6    L..    JMP  &C6AF
CD59 08          .      PHP
CD5A CD 65 03    .e.    CMP  &0365
CD5D F0 19       ..     BEQ  &CD78
CD5F 28          (      PLP
CD60 90 04       ..     BCC  &CD66
CD62 CE 65 03    .e.    DEC  &0365
CD65 60          `      RTS
CD66 EE 65 03    .e.    INC  &0365
CD69 60          `      RTS

CD6A 08          .      PHP    Save A, ST.
CD6B 48          H      PHA
CD6C AC 4F 03    .O.    LDY  &034F     No of bytes per screen char.
CD6F 88          .      DEY            -1
CD70 D0 1D       ..     BNE  &CD8F     branch if not Mode 7.
CD72 AD 38 03    .8.    LDA  &0338     Save on Screen
CD75 91 D8       ..     STA  (&D8),Y
CD77 68          h      PLA            Restore & exit
CD78 28          (      PLP
CD79 60          `      RTS

CD7A 08          .      PHP
CD7B 48          H      PHA
CD7C AC 4F 03    .O.    LDY  &034F
CD7F 88          .      DEY
CD80 D0 0D       ..     BNE  &CD8F
CD82 B1 D8       ..     LDA  (&D8),Y
```

```
CD84 8D 38 03     .8.     STA  &0338
CD87 AD 66 03     .f.     LDA  &0366
CD8A 91 D8        ..      STA  (&D8),Y
CD8C 4C 77 CD     Lw.     JMP  &CD77
CD8F A9 FF        ..      LDA  #&FF
CD91 C0 1F        ..      CPY  #&1F   } Branch if not Mode 2
CD93 D0 02        ..      BNE  &CD97
CD95 A9 3F        .?      LDA  #&3F
CD97 85 DA        ..      STA  &DA    &DA= &FF all modes except 7&2 else &3F
CD99 B1 D8        ..      LDA  (&D8),Y
CD9B 45 DA        E.      EOR  &DA
CD9D 91 D8        ..      STA  (&D8),Y
CD9F 88           .       DEY
CDA0 10 F7        ..      BPL  &CD99
CDA2 30 D3        0.      BMI  &CD77  exit, restoring registers
CDA4 20 5B CE      [.     JSR  &CE5B
CDA7 AD 09 03     ...     LDA  &0309
CDAA 8D 19 03     ...     STA  &0319
CDAD 20 06 CF      ..     JSR  &CF06
CDB0 20 F8 CC      ..     JSR  &CCF8
CDB3 B0 03        ..      BCS  &CDB8
CDB5 6D 54 03     mT.     ADC  &0354
CDB8 85 DB        ..      STA  &DB
CDBA 86 DA        ..      STX  &DA
CDBC 85 DC        ..      STA  &DC
CDBE B0 06        ..      BCS  &CDC6
CDC0 20 73 CE      s.     JSR  &CE73
CDC3 4C CE CD     L..     JMP  &CDCE
CDC6 20 F8 CC      ..     JSR  &CCF8
CDC9 90 F5        ..      BCC  &CDC0
CDCB 20 38 CE      8.     JSR  &CE38
CDCE A5 DC        ..      LDA  &DC
CDD0 A6 DA        ..      LDX  &DA
CDD2 85 D9        ..      STA  &D9
CDD4 86 D8        ..      STX  &D8
CDD6 C6 DE        ..      DEC  &DE
CDD8 D0 D6        ..      BNE  &CDB0
CDDA A2 28        .(      LDX  #&28
CDDC A0 18        ..      LDY  #&18
CDDE A9 02        ..      LDA  #&02   } Always branch.
CDE0 D0 06        ..      BNE  &CDE8
CDE2 A2 24        .$      LDX  #&24
CDE4 A0 14        ..      LDY  #&14
CDE6 A9 04        ..      LDA  #&04
CDE8 85 DA        ..      STA  &DA    temp workspace
CDEA BD 00 03     ...     LDA  &0300,X
CDED 48           H       PHA         save
CDEE B9 00 03     ...     LDA  &0300,Y  copy lst
CDF1 9D 00 03     ...     STA  &0300,X
CDF4 68           h       PLA         retrieve
CDF5 99 00 03     ...     STA  &0300,Y  copy 2nd.
CDF8 E8           .       INX  } increment offsets
CDF9 C8           .       INY
CDFA C6 DA        ..      DEC  &DA    decrement counter.
CDFC D0 EC        ..      BNE  &CDEA  branch if not complete.
CDFE 60           `       RTS  exit.
CDFF 20 5B CE      [.     JSR  &CE5B
CE02 AC 0B 03     ...     LDY  &030B
CE05 8C 19 03     ...     STY  &0319
CE08 20 06 CF      ..     JSR  &CF06
CE0B 20 D4 CA      ..     JSR  &CAD4
CE0E 10 04        ..      BPL  &CE14
```

Subroutine
Swap VDU

Variables
A= number to be swapped
X= offset of 1st set
Y= offset of 2nd set.

```
CE10 38          8      SEC
CE11 ED 54 03    .T.    SBC  &0354
CE14 85 DB       ..     STA  &DB
CE16 86 DA       ..     STX  &DA
CE18 85 DC       ..     STA  &DC
CE1A 90 06       ..     BCC  &CE22
CE1C 20 73 CE     s.    JSR  &CE73
CE1F 4C 2A CE    L*.    JMP  &CE2A
CE22 20 D4 CA     ..    JSR  &CAD4
CE25 30 F5       0.     BMI  &CE1C
CE27 20 38 CE     8.    JSR  &CE38
CE2A A5 DC       ..     LDA  &DC
CE2C A6 DA       ..     LDX  &DA
CE2E 85 D9       ..     STA  &D9
CE30 86 D8       ..     STX  &D8
CE32 C6 DE       ..     DEC  &DE
CE34 D0 D5       ..     BNE  &CE0B
CE36 F0 A2       ..     BEQ  &CDDA
CE38 AE 4D 03    .M.    LDX  &034D
CE3B F0 10       ..     BEQ  &CE4D
CE3D A0 00       ..     LDY  #&00
CE3F B1 DA       ..     LDA  (&DA),Y
CE41 91 D8       ..     STA  (&D8),Y
CE43 C8          .      INY
CE44 D0 F9       ..     BNE  &CE3F
CE46 E6 D9       ..     INC  &D9
CE48 E6 DB       ..     INC  &DB
CE4A CA          .      DEX
CE4B D0 F2       ..     BNE  &CE3F
CE4D AC 4C 03    .L.    LDY  &034C
CE50 F0 08       ..     BEQ  &CE5A
CE52 88          .      DEY
CE53 B1 DA       ..     LDA  (&DA),Y
CE55 91 D8       ..     STA  (&D8),Y
CE57 98          .      TYA
CE58 D0 F8       ..     BNE  &CE52
CE5A 60          `      RTS
CE5B 20 DA CD     ..    JSR  &CDDA
CE5E 38          8      SEC
CE5F AD 09 03    ...    LDA  &0309
CE62 ED 0B 03    ...    SBC  &030B
CE65 85 DE       ..     STA  &DE
CE67 D0 05       ..     BNE  &CE6E
CE69 68          h      PLA
CE6A 68          h      PLA
CE6B 4C DA CD    L..    JMP  &CDDA
CE6E AD 08 03    ...    LDA  &0308
CE71 10 70       .p     BPL  &CEE3
CE73 A5 DA       ..     LDA  &DA
CE75 48          H      PHA
CE76 38          8      SEC
CE77 AD 0A 03    ...    LDA  &030A
CE7A ED 08 03    ...    SBC  &0308
CE7D 85 DF       ..     STA  &DF
CE7F AC 4F 03    .O.    LDY  &034F
CE82 88          .      DEY
CE83 B1 DA       ..     LDA  (&DA),Y
CE85 91 D8       ..     STA  (&D8),Y
CE87 88          .      DEY
CE88 10 F9       ..     BPL  &CE83
CE8A A2 02       ..     LDX  #&02
CE8C 18          .      CLC
```

```
CE8D B5 D8      ..      LDA  &D8,X
CE8F 6D 4F 03   mO.     ADC  &034F
CE92 95 D8      ..      STA  &D8,X
CE94 B5 D9      ..      LDA  &D9,X
CE96 69 00      i.      ADC  #&00
CE98 10 04      ..      BPL  &CE9E
CE9A 38         8       SEC
CE9B ED 54 03   .T.     SBC  &0354
CE9E 95 D9      ..      STA  &D9,X
CEA0 CA         .       DEX
CEA1 CA         .       DEX
CEA2 F0 E8      ..      BEQ  &CE8C
CEA4 C6 DF      ..      DEC  &DF
CEA6 10 D7      ..      BPL  &CE7F
CEA8 68         h       PLA
CEA9 85 DA      ..      STA  &DA
CEAB 60         `       RTS
CEAC AD 18 03   ...     LDA  &0318
CEAF 48         H       PHA
CEB0 20 6E CE    n.     JSR  &CE6E
CEB3 20 06 CF    ..     JSR  &CF06
CEB6 38         8       SEC
CEB7 AD 0A 03   ...     LDA  &030A
CEBA ED 08 03   ...     SBC  &0308
CEBD 85 DC      ..      STA  &DC
CEBF AD 58 03   .X.     LDA  &0358
CEC2 AC 4F 03   .O.     LDY  &034F
CEC5 88         .       DEY
CEC6 91 D8      ..      STA  (&D8),Y
CEC8 D0 FB      ..      BNE  &CEC5
CECA 8A         .       TXA
CECB 18         .       CLC
CECC 6D 4F 03   mO.     ADC  &034F
CECF AA         .       TAX
CED0 A5 D9      ..      LDA  &D9
CED2 69 00      i.      ADC  #&00
CED4 10 04      ..      BPL  &CEDA
CED6 38         8       SEC
CED7 ED 54 03   .T.     SBC  &0354
CEDA 86 D8      ..      STX  &D8
CEDC 85 D9      ..      STA  &D9
CEDE C6 DC      ..      DEC  &DC
CEE0 10 DD      ..      BPL  &CEBF
CEE2 68         h       PLA
CEE3 8D 18 03   ...     STA  &0318
CEE6 38         8       SEC
CEE7 60         `       RTS
CEE8 AE 18 03   ...     LDX  &0318
CEEB EC 08 03   ...     CPX  &0308
CEEE 30 F6      0.      BMI  &CEE6
CEF0 EC 0A 03   ...     CPX  &030A
CEF3 F0 02      ..      BEQ  &CEF7
CEF5 10 EF      ..      BPL  &CEE6
CEF7 AE 19 03   ...     LDX  &0319
CEFA EC 0B 03   ...     CPX  &030B
CEFD 30 E7      0.      BMI  &CEE6
CEFF EC 09 03   ...     CPX  &0309
CF02 F0 02      ..      BEQ  &CF06
CF04 10 E0      ..      BPL  &CEE6
CF06 AD 19 03   ...     LDA  &0319
CF09 0A         .       ASL  A
CF0A A8         .       TAY
```

&CF06 calc Actual Screen Addre of text curs

current Y position of cursor

*2.

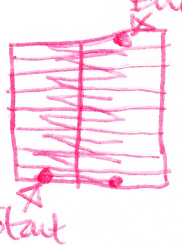
Y= index for &2280 table

```
CF0B B1 E0      ..      LDA  (&E0),Y   } Read high of screen address offset
CF0D 85 D9      ..      STA  &D9       }
CF0F C8         .       INY
CF10 A9 02      ..      LDA  #&02      Read memory map type & check the Modes 4 5 & 6 bit
CF12 2D 56 03   -V.     AND  &0356     (1= in mode 4,5 or 6)
CF15 08         .       PHP            Save ST
CF16 B1 E0      ..      LDA  (&E0),Y   Read low of screen address offset
CF18 28         (       PLP            Recover ST
CF19 F0 03      ..      BEQ  &CF1E     branch if in IBK modes (4,5 or 6)
CF1B 46 D9      F.      LSR  &D9       high /2
CF1D 6A         j       ROR  A         low /2
CF1E 6D 50 03   mP.     ADC  &0350     add low of screen start address
CF21 85 D8      ..      STA  &D8       Store low of address
CF23 A5 D9      ..      LDA  &D9
CF25 6D 51 03   mQ.     ADC  &0351     Y & A = High } Final screen address.
CF28 A8         .       TAY            }
CF29 AD 18 03   ...     LDA  &0318     Text cursor X co-ord
CF2C AE 4F 03   .O.     LDX  &034F     No of bytes used per screen character
CF2F CA         .       DEX            -1 = X
CF30 F0 12      ..      BEQ  &CF44     branch if in Mode 7
CF32 E0 0F      ..      CPX  #&0F
CF34 F0 03      ..      BEQ  &CF39     branch if [redacted] Mode 1,5.
CF36 90 02      ..      BCC  &CF3A     branch if Modes 0,3,4,6 (Drop through in Mode 2)
CF38 0A         .       ASL  A
CF39 0A         .       ASL  A         Xcord *16 (Mode2)   *8 (Mode 1,5)
CF3A 0A         .       ASL  A         *4 (Mode 0,3,4,6)
CF3B 0A         .       ASL  A
CF3C 90 02      ..      BCC  &CF40
CF3E C8         .       INY  } INC high of Final screen address twice
CF3F C8         .       INY  }
CF40 0A         .       ASL  A         X cord * 2 (All modes, downwards except
CF41 90 02      ..      BCC  &CF45
CF43 C8         .       INY            Inc high of Screen Addr.
CF44 18         .       CLC
CF45 65 D8      e.      ADC  &D8       Add to low of 'Saved' [redacted] Address
CF47 85 D8      ..      STA  &D8
CF49 8D 4A 03   .J.     STA  &034A     low of Scn as sent to 6845
CF4C AA         .       TAX            to X
CF4D 98         .       TYA            ACC = high of screen address.
CF4E 69 00      i.      ADC  #&00      (+ carry)
CF50 8D 4B 03   .K.     STA  &034B     Add extra to high of Scn Address
CF53 10 04      ..      BPL  &CF59     Branch if no wrap around
CF55 38         8       SEC
CF56 ED 54 03   .T.     SBC  &0354     } Subtract Screen size high byte
CF59 85 D9      ..      STA  &D9       Save high
CF5B 18         .       CLC
CF5C 60         `       RTS            Exit, C=0

CF5D AE 59 03   .Y.     LDX  &0359     Foreground Graphics colour
CF60 AC 5B 03   .[.     LDY  &035B     in plot mode (&DC) } VDU 5 Chara
CF63 20 B3 D0   ..      JSR  &D0B3     Calc colour bytes for graphics
CF66 20 86 D4   ..      JSR  &D486     Copy current co-ords to &328 (4 bytes)  Print routin
CF69 A0 00      ..      LDY  #&00      }
CF6B 84 DC      ..      STY  &DC       offset = 0 for char defn data
CF6D A4 DC      ..      LDY  &DC       }
CF6F B1 DE      .       LDA  (&DE),Y   read next byte of character defn.
CF71 F0 13      ..      BEQ  &CF86     = 0 SKIP
CF73 85 DD      .       STA  &DD       save byte in &DD
CF75 10 03      ..      BPL  &CF7A     }
CF77 20 E3 D0   ..      JSR  &D0E3     Display pixel if within graphics window.
CF7A EE 24 03   .$.     INC  &0324     Inc graphics cursor X Co-ord (Internal
CF7D D0 03      ..      BNE  &CF82     }
CF7F EE 25 03   .%.     INC  &0325     high of X co-ord.
```

```
CF82 06 DD        ..    ASL  &DD
CF84 D0 EF        ..    BNE  &CF75
CF86 A2 28        .(    LDX  #&28
CF88 A0 24        .$    LDY  #&24
CF8A 20 82 D4      ..   JSR  &D482
CF8D AC 26 03     .&.   LDY  &0326
CF90 D0 03        ..    BNE  &CF95
CF92 CE 27 03     .'.   DEC  &0327
CF95 CE 26 03     .&.   DEC  &0326
CF98 A4 DC        ..    LDY  &DC
CF9A C8           .     INY
CF9B C0 08        ..    CPY  #&08
CF9D D0 CC        ..    BNE  &CF6B
CF9F A2 28        .(    LDX  #&28
CFA1 A0 24        .$    LDY  #&24
CFA3 4C 8A D4     L..   JMP  &D48A
CFA6 A2 06        ..    LDX  #&06
CFA8 A0 26        .&    LDY  #&26
CFAA 20 82 D4      ..   JSR  &D482
CFAD A2 00        ..    LDX  #&00
CFAF A0 24        .$    LDY  #&24
CFB1 20 82 D4      ..   JSR  &D482
CFB4 4C B8 D1     L..   JMP  &D1B8
CFB7 AE 60 03     .`.   LDX  &0360
CFBA F0 20        .     BEQ  &CFDC
CFBC 20 3E D0      >.   JSR  &D03E
CFBF AE 60 03     .`.   LDX  &0360
CFC2 A5 D0        ..    LDA  &D0
CFC4 29 20        )     AND  #&20
CFC6 D0 95        ..    BNE  &CF5D
CFC8 A0 07        ..    LDY  #&07
CFCA E0 03        ..    CPX  #&03
CFCC F0 20        .     BEQ  &CFEE
CFCE B0 4E        .N    BCS  &D01E
CFD0 B1 DE        ..    LDA  (&DE),Y
CFD2 05 D2        ..    ORA  &D2
CFD4 45 D3        E.    EOR  &D3
CFD6 91 D8        ..    STA  (&D8),Y
CFD8 88           .     DEY
CFD9 10 F5        ..    BPL  &CFD0
CFDB 60           `     RTS
CFDC A0 02        ..    LDY  #&02
CFDE D9 B6 C4     ...   CMP  &C4B6,Y
CFE1 F0 06        ..    BEQ  &CFE9
CFE3 88           .     DEY
CFE4 10 F8        ..    BPL  &CFDE
CFE6 81 D8        ..    STA  (&D8,X)
CFE8 60           `     RTS
CFE9 B9 B7 C4     ...   LDA  &C4B7,Y
CFEC D0 F8        ..    BNE  &CFE6
CFEE B1 DE        ..    LDA  (&DE),Y
CFF0 48           H     PHA
CFF1 4A           J     LSR  A
CFF2 4A           J     LSR  A
CFF3 4A           J     LSR  A
CFF4 4A           J     LSR  A
CFF5 AA           .     TAX
CFF6 BD 1F C3     ...   LDA  &C31F,X
CFF9 05 D2        ..    ORA  &D2
CFFB 45 D3        E.    EOR  &D3
CFFD 91 D8        ..    STA  (&D8),Y
CFFF 98           .     TYA
```

get next bit of character defn in 'C'. '28
branch ~~if 0~~ char. def byte Ø.
Copy 2 bytes from &328 to &324
'Y' low graphics cursor (internal)
'Y' coord = Y coord - 1 (internal)
All 8 bytes of defn. done? branch if not
Restore original (top left) position of graphics cursor and exit VDU 5.
no of colours in current mode (-1).
branch if no colours (mode 7 only)
get low & high byte of character definition
no of colours in mode.
VDU status.
VDU 5 on
branch if VDU 5 enabled.
Y=7
is it a 4 colour mode
branch if it is a 4 colour mode (1 & 5)
branch if less than 4 colours (Ø, 3, 4, 6)
text colour bytes to be ORed with memory
" " " " " EORed " "
top left of character cell.
} go up toward top left of cell
exit.
} check char
X is always Ø.
Teletext conversion.
Teletext character print.
convert.
4 colour mode char print.
Y=7 on entry!
X = upper nibble of char def.
ADD 8 to offset



```
?D D000 F000
D000 18          .      CLC
D001 69 08       i.     ADC  #&08
D003 A8          .      TAY
D004 68          h      PLA
D005 29 0F       ).     AND  #&0F
D007 AA          .      TAX
D008 BD 1F C3    ...    LDA  &C31F,X
D00B 05 D2       ..     ORA  &D2
D00D 45 D3       E.     EOR  &D3
D00F 91 D8       ..     STA  (&D8),Y
D011 98          .      TYA
D012 E9 08       ..     SBC  #&08
D014 A8          .      TAY
D015 10 D7       ..     BPL  &CFEE
D017 60          `      RTS
D018 98          .      TYA
D019 E9 21       .!     SBC  #&21
D01B 30 FA       0.     BMI  &D017
D01D A8          .      TAY
D01E B1 DE       ..     LDA  (&DE),Y
D020 85 DC       ..     STA  &DC
D022 38          8      SEC
D023 A9 00       ..     LDA  #&00
D025 26 DC       &.     ROL  &DC
D027 F0 EF       ..     BEQ  &D018
D029 2A          *      ROL  A
D02A 06 DC       ..     ASL  &DC
D02C 2A          *      ROL  A
D02D AA          .      TAX
D02E BD 2F C3    ./.    LDA  &C32F,X
D031 05 D2       ..     ORA  &D2
D033 45 D3       E.     EOR  &D3
D035 91 D8       ..     STA  (&D8),Y
D037 18          .      CLC
D038 98          .      TYA
D039 69 08       i.     ADC  #&08
D03B A8          .      TAY
D03C 90 E5       ..     BCC  &D023
D03E 0A          .      ASL  A    *2
D03F 2A          *      ROL  A    *2
D040 2A          *      ROL  A    *2
D041 85 DE       ..     STA  &DE
D043 29 03       ).     AND  #&03
D045 2A          *      ROL  A    *2
D046 AA          .      TAX
D047 29 03       ).     AND  #&03
D049 69 BF       i.     ADC  #&BF
D04B A8          .      TAY
D04C BD 0D C4    ...    LDA  &C40D,X
D04F 2C 67 03    ,g.    BIT  &0367
D052 F0 03       ..     BEQ  &D057
D054 BC 67 03    .g.    LDY  &0367,X
D057 84 DF       ..     STY  &DF
D059 A5 DE       ..     LDA  &DE
D05B 29 F8       ).     AND  #&F8
D05D 85 DE       ..     STA  &DE
D05F 60          `      RTS
D060 A2 20       .      LDX  #&20
D062 20 4D D1     M.    JSR  &D14D
D065 AD 1F 03    ...    LDA  &031F
```

add 8 to offset.

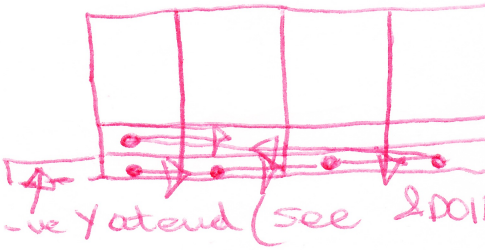
X= lower nibble of char def

Subtract 9 from offset (C=0 always)

exit

(Y=7 on entry) 16 colour Mode character

8 to step &D027 being taken during first loop. Print




ADD 8 to offset.

always taken!

A=character being printed,

temp workspace (low byte of character address)

(bit 7&6 of original char

X=result. (bit 7&6 *2)

add &BF (C is always 0) to give &C0 to &

Y=result.

mask table for 2 colour mode. (X=bit mul

Font flag

branch if Font ID From ROM.

read font location byte (upper byte).

Store high at &DF (of character address

remove bits 0-2 (get start of character

definition

exit.

Put Final coords into &324 to &327

Main PLOT entry point

The current PLOT number.

```
D068 C9 04        ..      CMP  #&04
D06A F0 6D        .m      BEQ  &DOD9
D06C A0 05        ..      LDY  #&05
D06E 29 03        ).      AND  #&03
D070 F0 0E        ..      BEQ  &D080
D072 4A           J       LSR  A
D073 B0 03        ..      BCS  &D078
D075 88           .       DEY
D076 D0 08        ..      BNE  &D080
D078 AA           .       TAX
D079 BC 5B 03     .[.     LDY  &035B,X
D07C BD 59 03     .Y.     LDA  &0359,X
D07F AA           .       TAX
D080 20 B3 D0      ..     JSR  &DOB3
D083 AD 1F 03     ...     LDA  &031F
D086 30 23        0#      BMI  &DOAB
D088 0A           .       ASL  A
D089 10 3B        .;      BPL  &DOC6
D08B 29 F0        ).      AND  #&F0
D08D 0A           .       ASL  A
D08E F0 46        .F      BEQ  &DOD6
D090 49 40        I@      EOR  #&40
D092 F0 14        ..      BEQ  &DOA8
D094 48           H       PHA
D095 20 DC D0      ..     JSR  &DODC
D098 68           h       PLA
D099 49 60        I`      EOR  #&60
D09B F0 11        ..      BEQ  &DOAE
D09D C9 40        .@      CMP  #&40
D09F D0 0A        ..      BNE  &DOAB
DOA1 A9 02        ..      LDA  #&02
DOA3 85 DC        ..      STA  &DC
DOA5 4C 06 D5     L..     JMP  &D506
DOA8 4C EA D5     L..     JMP  &D5EA
DOAB 4C 38 C9     L8.     JMP  &C938
DOAE 85 DC        ..      STA  &DC
DOB0 4C BF D4     L..     JMP  &D4BF
DOB3 8A           .       TXA
DOB4 19 1C C4     ...     ORA  &C41C,Y
DOB7 59 1D C4     Y..     EOR  &C41D,Y
DOBA 85 D4        ..      STA  &D4
DOBC 8A           .       TXA
DOBD 19 1B C4     ...     ORA  &C41B,Y
DOC0 59 20 C4     Y .     EOR  &C420,Y
DOC3 85 D5        ..      STA  &D5
DOC5 60           `       RTS
DOC6 0A           .       ASL  A
DOC7 30 E2        0.      BMI  &DOAB
DOC9 0A           .       ASL  A
DOCA 0A           .       ASL  A
DOCB 10 03        ..      BPL  &DOD0
DOCD 20 EB D0      ..     JSR  &DOEB
DOD0 20 ED D1      ..     JSR  &D1ED
DOD3 4C D9 D0     L..     JMP  &DOD9
DOD6 20 EB D0      ..     JSR  &DOEB
DOD9 20 E2 CD      ..     JSR  &CDE2
DODC A0 24        .$      LDY  #&24
DODE A2 20        .       LDX  #&20
DOE0 4C 8A D4     L..     JMP  &D48A
DOE3 A2 24        .$      LDX  #&24
DOE5 20 5F D8      _.     JSR  &D85F
DOE8 F0 06        ..      BEQ  &DOF0
```

Branch if PLOT 4 (Move absolute) exit after

Branch only if PLOT option is MOVE (relative or Absolute)

Branch if drawing in foreground colour

Put colour bytes in &D4, &D5

Branch if -ve plot option (128 → 255

Branch if PLOT between 0 and 63

A = 2 * (PLOT * 2 AND &F0)

∴ PLOT between 64 & 127

Branch if PLOT 0 → 63?

Branch if PLOT 16 → 23?

Branch if not recognised PLOT.

&DC = 2

For negative PLOT option / PLOT (32 – 63) y/inc

work out graphics colour bytes

graphics colour bytes

Branch if PLOT between 32 and 63

Branch if PLOT 0 → 7 inclusive

PLOT 8 → 31 (?)

Swap &324 & &314 (4 bytes)

Copy 4 bytes from &320 → &324

result in &D6, &D7

Y = offset

in X

Calculation of pixel using 4 bytes at &32

Branch if pixel was inside window.

```
D0EA 60          `      RTS     exit if window violation
D0EB 20 5D D8     ].    JSR  &D85D
D0EE D0 13        ..    BNE  &D103
D0F0 AC 1A 03     ...   LDY  &031A     - not needed!?
D0F3 A5 D1        ..    LDA  &D1       mask for pixelposition
D0F5 25 D4        %.    AND  &D4       " " colour
D0F7 11 D6        ..    ORA  (&D6),Y   OR with screen
D0F9 85 DA        ..    STA  &DA       Save (temp)
D0FB A5 D5        ..    LDA  &D5       mask for colour
D0FD 25 D1        %.    AND  &D1       " " pixel position
D0FF 45 DA        E.    EOR  &DA       EOR with 1st result.
D101 91 D6        ..    STA  (&D6),Y   SAVE to screen.
D103 60          `      RTS
D104 B1 D6        ..    LDA  (&D6),Y
D106 05 D4        ..    ORA  &D4
D108 45 D5        E.    EOR  &D5
D10A 91 D6        ..    STA  (&D6),Y
D10C 60          `      RTS
D10D A2 24        .$    LDX  #&24
D10F A0 00        ..    LDY  #&00      &DA=0
D111 84 DA        ..    STY  &DA
D113 A0 02        ..    LDY  #&02
D115 20 28 D1      (.   JSR  &D128     Window check
D118 06 DA        ..    ASL  &DA
D11A 06 DA        ..    ASL  &DA
D11C CA           .     DEX  } Decrement offset
D11D CA           .     DEX  }
D11E A0 00        ..    LDY  #&00
D120 20 28 D1      (.   JSR  &D128     Window check.
D123 E8           .     INX  } Restore offset to original value
D124 E8           .     INX  }
D125 A5 DA        ..    LDA  &DA       &DA=0 if No violation of Window
D127 60          `      RTS

D128 BD 02 03     ...   LDA  &0302,X   Window violation
D12B D9 00 03     ...   CMP  &0300,Y   checking routine
D12E BD 03 03     ...   LDA  &0303,X   (INC &DA) executed
D131 F9 01 03     ...   SBC  &0301,Y   if TRUE.
D134 30 10        0.    BMI  &D146
D136 B9 04 03     ...   LDA  &0304,Y
D139 DD 02 03     ...   CMP  &0302,X
D13C B9 05 03     ...   LDA  &0305,Y
D13F FD 03 03     ...   SBC  &0303,X
D142 10 04        ..    BPL  &D148
D144 E6 DA        ..    INC  &DA
D146 E6 DA        ..    INC  &DA
D148 60          `      RTS

D149 A9 FF        ..    LDA  #&FF
D14B D0 03        ..    BNE  &D150
D14D AD 1F 03     ...   LDA  &031F     &DA= the current PLOT number.
D150 85 DA        ..    STA  &DA
D152 A0 02        ..    LDY  #&02      Put Final point Y coordinate into &312,& 313.
D154 20 76 D1      v.   JSR  &D176     "Divide" Y ("326,327") + graphicsorigin into &326,&327
D157 20 AD D1      ..   JSR  &D1AD     Divide Y (&326,&327) by 2
D15A A0 00        ..    LDY  #&00      Y=0
D15C CA           .     DEX  } X now &22.
D15D CA           .     DEX  }
D15E 20 76 D1      v.   JSR  &D176     Put Final 'X' into &310,&311 + graphicorigin in &324,
D161 AC 61 03     .a.   LDY  &0361
D164 C0 03        ..    CPY  #&03      Branch if No of pixels per byte=4(-1) ie, Modes 1,5
D166 F0 05        ..    BEQ  &D16D
D168 B0 06        ..    BCS  &D170     Branch if two colour mode ie, Modes 0,3,4,6
D16A 20 AD D1      ..   JSR  &D1AD     Divide 'X' by 2 (&324,&325)
```

```
D16D 20 AD D1      ..    JSR  &D1AD
D170 AD 56 03      .V.   LDA  &0356
D173 D0 38         .8    BNE  &D1AD
D175 60            `     RTS
D176 18            .     CLC
D177 A5 DA         ..    LDA  &DA
D179 29 04         ).    AND  #&04
D17B F0 09         ..    BEQ  &D186
D17D BD 02 03      ...   LDA  &0302,X
D180 48            H     PHA
D181 BD 03 03      ...   LDA  &0303,X
D184 90 0E         ..    BCC  &D194
D186 BD 02 03      ...   LDA  &0302,X
D189 79 10 03      y..   ADC  &0310,Y
D18C 48            H     PHA
D18D BD 03 03      ...   LDA  &0303,X
D190 79 11 03      y..   ADC  &0311,Y
D193 18            .     CLC
D194 99 11 03      ...   STA  &0311,Y
D197 79 0D 03      y..   ADC  &030D,Y
D19A 9D 03 03      ...   STA  &0303,X
D19D 68            h     PLA
D19E 99 10 03      ...   STA  &0310,Y
D1A1 18            .     CLC
D1A2 79 0C 03      y..   ADC  &030C,Y
D1A5 9D 02 03      ...   STA  &0302,X
D1A8 90 03         ..    BCC  &D1AD
D1AA FE 03 03      ...   INC  &0303,X
D1AD BD 03 03      ...   LDA  &0303,X
D1B0 0A            .     ASL  A
D1B1 7E 03 03      ~..   ROR  &0303,X
D1B4 7E 02 03      ~..   ROR  &0302,X
D1B7 60            `     RTS
D1B8 A0 10         ..    LDY  #&10
D1BA 20 88 D4       ..   JSR  &D488
D1BD A2 02         ..    LDX  #&02
D1BF A0 02         ..    LDY  #&02
D1C1 20 D5 D1       ..   JSR  &D1D5
D1C4 A2 00         ..    LDX  #&00
D1C6 A0 04         ..    LDY  #&04
D1C8 AD 61 03      .a.   LDA  &0361
D1CB 88            .     DEY
D1CC 4A            J     LSR  A
D1CD D0 FC         ..    BNE  &D1CB
D1CF AD 56 03      .V.   LDA  &0356
D1D2 F0 01         ..    BEQ  &D1D5
D1D4 C8            .     INY
D1D5 1E 10 03      ...   ASL  &0310,X
D1D8 3E 11 03      >..   ROL  &0311,X
D1DB 88            .     DEY
D1DC D0 F7         ..    BNE  &D1D5
D1DE 38            8     SEC
D1DF 20 E3 D1       ..   JSR  &D1E3
D1E2 E8            .     INX
D1E3 BD 10 03      ...   LDA  &0310,X
D1E6 FD 0C 03      ...   SBC  &030C,X
D1E9 9D 10 03      ...   STA  &0310,X
D1EC 60            `     RTS
D1ED 20 0D D4       ..   JSR  &D40D
D1F0 AD 2B 03      .+.   LDA  &032B
D1F3 4D 29 03      M).   EOR  &0329
D1F6 30 0F         0.    BMI  &D207
```

Divide final X position by 2 (8324, 8325)

Branch if memory map type <>0 (ie all modes except 2)

exit

(divide by 2 again) X=824, Y=2

c=0

A= current PLOT option

branch if relative PLOT type selected.

read current Y position of graphics cursor.

always taken.

Calculate new 'relative' position of Y coord

Adjust using graphics coordinates of the origin (Y coordinates)

INC high byte

*2

Divide pair by 2

Find difference between the 2 sets of coordinates.

PLOT 0 to 7

DX high

branch if -ve

```
D1F8 AD 2A 03     .*.    LDA  &032A
D1FB CD 28 03     .(.    CMP  &0328
D1FE AD 2B 03     .+.    LDA  &032B
D201 ED 29 03     .).    SBC  &0329
D204 4C 14 D2     L..    JMP  &D214
D207 AD 28 03     .(.    LDA  &0328
D20A 18           .      CLC
D20B 6D 2A 03     m*.    ADC  &032A
D20E AD 29 03     .).    LDA  &0329
D211 6D 2B 03     m+.    ADC  &032B
D214 6A           j      ROR  A
D215 A2 00        ..     LDX  #&00
D217 4D 2B 03     M+.    EOR  &032B
D21A 10 02        ..     BPL  &D21E
D21C A2 02        ..     LDX  #&02
D21E 86 DE        ..     STX  &DE
D220 BD AA C4     ...    LDA  &C4AA,X
D223 8D 5D 03     .].    STA  &035D
D226 BD AB C4     ...    LDA  &C4AB,X
D229 8D 5E 03     .^.    STA  &035E
D22C BD 29 03     .).    LDA  &0329,X
D22F 10 04        ..     BPL  &D235
D231 A2 24        .$     LDX  #&24
D233 D0 02        ..     BNE  &D237
D235 A2 20        .      LDX  #&20
D237 86 DF        ..     STX  &DF
D239 A0 2C        .,     LDY  #&2C
D23B 20 8A D4       ..   JSR  &D48A
D23E A5 DF        ..     LDA  &DF
D240 49 04        I.     EOR  #&04
D242 85 DD        ..     STA  &DD
D244 05 DE        ..     ORA  &DE
D246 AA           .      TAX
D247 20 80 D4       ..   JSR  &D480
D24A AD 1F 03     ...    LDA  &031F
D24D 29 10        ).     AND  #&10
D24F 0A           .      ASL  A
D250 0A           .      ASL  A
D251 0A           .      ASL  A
D252 85 DB        ..     STA  &DB
D254 A2 2C        .,     LDX  #&2C
D256 20 0F D1       ..   JSR  &D10F
D259 85 DC        ..     STA  &DC
D25B F0 06        ..     BEQ  &D263
D25D A9 40        .@     LDA  #&40
D25F 05 DB        ..     ORA  &DB
D261 85 DB        ..     STA  &DB
D263 A6 DD        ..     LDX  &DD
D265 20 0F D1       ..   JSR  &D10F
D268 24 DC        $.     BIT  &DC
D26A F0 01        ..     BEQ  &D26D
D26C 60           `      RTS
D26D A6 DE        ..     LDX  &DE
D26F F0 02        ..     BEQ  &D273
D271 4A           J      LSR  A
D272 4A           J      LSR  A
D273 29 02        ).     AND  #&02
D275 F0 07        ..     BEQ  &D27E
D277 8A           .      TXA
D278 09 04        ..     ORA  #&04
D27A AA           .      TAX
D27B 20 80 D4       ..   JSR  &D480
```

DY low

DX low

DY high

DX high

DX low

DY low

DX high

DY high.

/2

DY high

&DE = 2 or 0

VDU link = &D386 for X=0
&D37E for X=2

DX high or DY high

&DF = &20 or &24

Copy 4 bytes from &320 or &324 to &32C

&DD = &24 or &20

X = &26, &24 or &20, &22

Copy &2 bytes to &330 (using X as source offset into &

PLOT number

Test bit 4

*8

? &DB = 0 or &80 ( ie, Bit 7 set if PLOT 16→23 (8)

check for window violation

&DC = 0 if inside window

Branch if OK.

? &DB = ?&DB OR &40.

check window violation

Branch if no violation

Can't go any further so return.

```
D27E 20 2C D4      ,.     JSR  &D42C
D281 A5 DE         ..     LDA  &DE
D283 49 02         I.     EOR  #&02
D285 AA            .      TAX
D286 A8            .      TAY
D287 AD 29 03      .).    LDA  &0329
D28A 4D 2B 03      M+.    EOR  &032B
D28D 10 01         ..     BPL  &D290
D28F E8            .      INX
D290 BD AE C4      ...    LDA  &C4AE,X
D293 8D 32 03      .2.    STA  &0332
D296 BD B2 C4      ...    LDA  &C4B2,X
D299 8D 33 03      .3.    STA  &0333
D29C A9 7F         ..     LDA  #&7F
D29E 8D 34 03      .4.    STA  &0334
D2A1 24 DB         $.     BIT  &DB
D2A3 70 29         p)     BVS  &D2CE
D2A5 BD 47 C4      .G.    LDA  &C447,X
D2A8 AA            .      TAX
D2A9 38            8      SEC
D2AA BD 00 03      ...    LDA  &0300,X
D2AD F9 2C 03      .,.    SBC  &032C,Y
D2B0 85 DA         ..     STA  &DA
D2B2 BD 01 03      ...    LDA  &0301,X
D2B5 F9 2D 03      .-.    SBC  &032D,Y
D2B8 A4 DA         ..     LDY  &DA
D2BA AA            .      TAX
D2BB 10 03         ..     BPL  &D2C0
D2BD 20 9B D4       ..    JSR  &D49B
D2C0 AA            .      TAX
D2C1 C8            .      INY
D2C2 D0 01         ..     BNE  &D2C5
D2C4 E8            .      INX
D2C5 8A            .      TXA
D2C6 F0 02         ..     BEQ  &D2CA
D2C8 A0 00         ..     LDY  #&00
D2CA 84 DF         ..     STY  &DF
D2CC F0 09         ..     BEQ  &D2D7
D2CE 8A            .      TXA
D2CF 4A            J      LSR  A
D2D0 6A            j      ROR  A
D2D1 09 02         ..     ORA  #&02
D2D3 45 DE         E.     EOR  &DE
D2D5 85 DE         ..     STA  &DE
D2D7 A2 2C         .,     LDX  #&2C
D2D9 20 64 D8       d.    JSR  &D864
D2DC A6 DC         ..     LDX  &DC
D2DE D0 02         ..     BNE  &D2E2
D2E0 C6 DD         ..     DEC  &DD
D2E2 CA            .      DEX
D2E3 A5 DB         ..     LDA  &DB
D2E5 F0 1F         ..     BEQ  &D306
D2E7 10 10         ..     BPL  &D2F9
D2E9 2C 34 03      ,4.    BIT  &0334
D2EC 10 05         ..     BPL  &D2F3
D2EE CE 34 03      .4.    DEC  &0334
D2F1 D0 23         .#     BNE  &D316
D2F3 EE 34 03      .4.    INC  &0334
D2F6 0A            .      ASL  A
D2F7 10 0D         ..     BPL  &D306
D2F9 86 DC         ..     STX  &DC
D2FB A2 2C         .,     LDX  #&2C
```

```
D2FD 20 5F D8      _.     JSR  &D85F
D300 A6 DC         ..     LDX  &DC
D302 09 00         ..     ORA  #&00
D304 D0 10         ..     BNE  &D316
D306 A5 D1         ..     LDA  &D1
D308 25 D4         %.     AND  &D4
D30A 11 D6         ..     ORA  (&D6),Y
D30C 85 DA         ..     STA  &DA
D30E A5 D5         ..     LDA  &D5
D310 25 D1         %.     AND  &D1
D312 45 DA         E.     EOR  &DA
D314 91 D6         ..     STA  (&D6),Y
D316 38            8      SEC
D317 AD 35 03      .5.    LDA  &0335
D31A ED 37 03      .7.    SBC  &0337
D31D 8D 35 03      .5.    STA  &0335
D320 AD 36 03      .6.    LDA  &0336
D323 ED 38 03      .8.    SBC  &0338
D326 B0 11         ..     BCS  &D339
D328 85 DA         ..     STA  &DA
D32A AD 35 03      .5.    LDA  &0335
D32D 6D 39 03      m9.    ADC  &0339
D330 8D 35 03      .5.    STA  &0335
D333 A5 DA         ..     LDA  &DA
D335 6D 3A 03      m:.    ADC  &033A
D338 18            .      CLC
D339 8D 36 03      .6.    STA  &0336
D33C 08            .      PHP
D33D B0 09         ..     BCS  &D348
D33F 6C 32 03      l2.    JMP  (&0332)
D342 88            .      DEY
D343 10 03         ..     BPL  &D348
D345 20 D3 D3       ..    JSR  &D3D3
D348 6C 5D 03      l].    JMP  (&035D)
D34B C8            .      INY
D34C C0 08         ..     CPY  #&08
D34E D0 F8         ..     BNE  &D348
D350 18            .      CLC
D351 A5 D6         ..     LDA  &D6
D353 6D 52 03      mR.    ADC  &0352
D356 85 D6         ..     STA  &D6
D358 A5 D7         ..     LDA  &D7
D35A 6D 53 03      mS.    ADC  &0353
D35D 10 04         ..     BPL  &D363
D35F 38            8      SEC
D360 ED 54 03      .T.    SBC  &0354
D363 85 D7         ..     STA  &D7
D365 A0 00         ..     LDY  #&00
D367 6C 5D 03      l].    JMP  (&035D)
D36A 46 D1         F.     LSR  &D1
D36C 90 DA         ..     BCC  &D348
D36E 20 ED D3       ..    JSR  &D3ED
D371 6C 5D 03      l].    JMP  (&035D)
D374 06 D1         ..     ASL  &D1
D376 90 D0         ..     BCC  &D348
D378 20 FD D3       ..    JSR  &D3FD
D37B 6C 5D 03      l].    JMP  (&035D)
D37E 88            .      DEY
D37F 10 0C         ..     BPL  &D38D
D381 20 D3 D3       ..    JSR  &D3D3
D384 D0 07         ..     BNE  &D38D
D386 46 D1         F.     LSR  &D1
```

```
D388 90 03        ..      BCC  &D38D
D38A 20 ED D3      ..     JSR  &D3ED
D38D 28           (       PLP
D38E E8           .       INX
D38F D0 04        ..      BNE  &D395
D391 E6 DD        ..      INC  &DD
D393 F0 0A        ..      BEQ  &D39F
D395 24 DB        $.      BIT  &DB
D397 70 07        p.      BVS  &D3A0
D399 B0 35        .5      BCS  &D3D0
D39B C6 DF        ..      DEC  &DF
D39D D0 31        .1      BNE  &D3D0
D39F 60           `       RTS
D3A0 A5 DE        ..      LDA  &DE
D3A2 86 DC        ..      STX  &DC
D3A4 29 02        ).      AND  #&02
D3A6 AA           .       TAX
D3A7 B0 19        ..      BCS  &D3C2
D3A9 24 DE        $.      BIT  &DE
D3AB 30 0A        0.      BMI  &D3B7
D3AD FE 2C 03     .,.     INC  &032C,X
D3B0 D0 10        ..      BNE  &D3C2
D3B2 FE 2D 03     .-.     INC  &032D,X
D3B5 90 0B        ..      BCC  &D3C2
D3B7 BD 2C 03     .,.     LDA  &032C,X
D3BA D0 03        ..      BNE  &D3BF
D3BC DE 2D 03     .-.     DEC  &032D,X
D3BF DE 2C 03     .,.     DEC  &032C,X
D3C2 8A           .       TXA
D3C3 49 02        I.      EOR  #&02
D3C5 AA           .       TAX
D3C6 FE 2C 03     .,.     INC  &032C,X
D3C9 D0 03        ..      BNE  &D3CE
D3CB FE 2D 03     .-.     INC  &032D,X
D3CE A6 DC        ..      LDX  &DC
D3D0 4C E3 D2     L..     JMP  &D2E3
D3D3 38           8       SEC
D3D4 A5 D6        ..      LDA  &D6
D3D6 ED 52 03     .R.     SBC  &0352
D3D9 85 D6        ..      STA  &D6
D3DB A5 D7        ..      LDA  &D7
D3DD ED 53 03     .S.     SBC  &0353
D3E0 CD 4E 03     .N.     CMP  &034E
D3E3 B0 03        ..      BCS  &D3E8
D3E5 6D 54 03     mT.     ADC  &0354
D3E8 85 D7        ..      STA  &D7
D3EA A0 07        ..      LDY  #&07
D3EC 60           `       RTS
D3ED AD 62 03     .b.     LDA  &0362
D3F0 85 D1        ..      STA  &D1
D3F2 A5 D6        ..      LDA  &D6
D3F4 69 07        i.      ADC  #&07
D3F6 85 D6        ..      STA  &D6
D3F8 90 02        ..      BCC  &D3FC
D3FA E6 D7        ..      INC  &D7
D3FC 60           `       RTS
D3FD AD 63 03     .c.     LDA  &0363
D400 85 D1        ..      STA  &D1
D402 A5 D6        ..      LDA  &D6
D404 D0 02        ..      BNE  &D408
D406 C6 D7        ..      DEC  &D7
D408 E9 08        ..      SBC  #&08
```

```
D40A 85 D6        ..      STA  &D6
D40C 60           `       RTS
D40D A0 28        .(      LDY  #&28
D40F A2 20        .       LDX  #&20
D411 20 18 D4      ..     JSR  &D418
D414 E8           .       INX
D415 E8           .       INX
D416 C8           .       INY
D417 C8           .       INY
D418 38           8       SEC
D419 BD 04 03     ...     LDA  &0304,X
D41C FD 00 03     ...     SBC  &0300,X
D41F 99 00 03     ...     STA  &0300,Y
D422 BD 05 03     ...     LDA  &0305,X
D425 FD 01 03     ...     SBC  &0301,X
D428 99 01 03     ...     STA  &0301,Y
D42B 60           `       RTS
D42C A5 DE        ..      LDA  &DE
D42E D0 07        ..      BNE  &D437
D430 A2 28        .(      LDX  #&28
D432 A0 2A        .*      LDY  #&2A
D434 20 DE CD      ..     JSR  &CDDE
D437 A2 28        .(      LDX  #&28
D439 A0 37        .7      LDY  #&37
D43B 20 8A D4      ..     JSR  &D48A
D43E 38           8       SEC
D43F A6 DE        ..      LDX  &DE
D441 AD 30 03     .0.     LDA  &0330
D444 FD 2C 03     .,.     SBC  &032C,X
D447 A8           .       TAY
D448 AD 31 03     .1.     LDA  &0331
D44B FD 2D 03     .-.     SBC  &032D,X
D44E 30 03        0.      BMI  &D453
D450 20 9B D4      ..     JSR  &D49B
D453 85 DD        ..      STA  &DD
D455 84 DC        ..      STY  &DC
D457 A2 35        .5      LDX  #&35
D459 20 67 D4      g.     JSR  &D467
D45C 4A           J       LSR  A
D45D 9D 01 03     ...     STA  &0301,X
D460 98           .       TYA
D461 6A           j       ROR  A
D462 9D 00 03     ...     STA  &0300,X
D465 CA           .       DEX
D466 CA           .       DEX
D467 BC 04 03     ...     LDY  &0304,X
D46A BD 05 03     ...     LDA  &0305,X
D46D 10 0C        ..      BPL  &D47B
D46F 20 9B D4      ..     JSR  &D49B
D472 9D 05 03     ...     STA  &0305,X
D475 48           H       PHA
D476 98           .       TYA
D477 9D 04 03     ...     STA  &0304,X
D47A 68           h       PLA
D47B 60           `       RTS
D47C A9 08        ..      LDA  #&08
D47E D0 0C        ..      BNE  &D48C
D480 A0 30        .0      LDY  #&30
D482 A9 02        ..      LDA  #&02
D484 D0 06        ..      BNE  &D48C
D486 A0 28        .(      LDY  #&28
D488 A2 24        .$      LDX  #&24
```

Find difference between X and Y coords & save in &328 → &32B.

Next pair of co ords

Find difference between 2 sets of co-ordinates.

} Copy 4 bytes from (current co-ord) &324 → &327 (previous coord) &328 → &32B

```
D48A A9 04        ..      LDA  #&04
D48C 85 DA        ..      STA  &DA
D48E BD 00 03     ...     LDA  &0300,X
D491 99 00 03     ...     STA  &0300,Y
D494 E8           .       INX
D495 C8           .       INY
D496 C6 DA        ..      DEC  &DA
D498 D0 F4        ..      BNE  &D48E
D49A 60           `       RTS
```

Copy ~~copy~~ 4 bytes elsewhere in page 3.

```
D49B 48           H       PHA
D49C 98           .       TYA
D49D 49 FF        I.      EOR  #&FF
D49F A8           .       TAY
D4A0 68           h       PLA
D4A1 49 FF        I.      EOR  #&FF
D4A3 C8           .       INY
D4A4 D0 03        ..      BNE  &D4A9
D4A6 18           .       CLC
D4A7 69 01        i.      ADC  #&01
D4A9 60           `       RTS
D4AA 20 5D D8      ].     JSR  &D85D
D4AD D0 08        ..      BNE  &D4B7
D4AF B1 D6        ..      LDA  (&D6),Y
D4B1 4D 5A 03     MZ.     EOR  &035A
D4B4 85 DA        ..      STA  &DA
D4B6 60           `       RTS
D4B7 68           h       PLA
D4B8 68           h       PLA
D4B9 EE 26 03     .&.     INC  &0326
D4BC 4C 45 D5     LE.     JMP  &D545
D4BF 20 AA D4      ..     JSR  &D4AA
D4C2 25 D1        %.      AND  &D1
D4C4 D0 F3        ..      BNE  &D4B9
D4C6 A2 00        ..      LDX  #&00
D4C8 20 92 D5      ..     JSR  &D592
D4CB F0 2D        .-      BEQ  &D4FA
D4CD AC 1A 03     ...     LDY  &031A
D4D0 06 D1        ..      ASL  &D1
D4D2 B0 05        ..      BCS  &D4D9
D4D4 20 74 D5      t.     JSR  &D574
D4D7 90 21        .!      BCC  &D4FA
D4D9 20 FD D3      ..     JSR  &D3FD
D4DC B1 D6        ..      LDA  (&D6),Y
D4DE 4D 5A 03     MZ.     EOR  &035A
D4E1 85 DA        ..      STA  &DA
D4E3 D0 12        ..      BNE  &D4F7
D4E5 38           8       SEC
D4E6 8A           .       TXA
D4E7 6D 61 03     ma.     ADC  &0361
D4EA 90 04        ..      BCC  &D4F0
D4EC E6 DB        ..      INC  &DB
D4EE 10 07        ..      BPL  &D4F7
D4F0 AA           .       TAX
D4F1 20 04 D1      ..     JSR  &D104
D4F4 38           8       SEC
D4F5 B0 E2        ..      BCS  &D4D9
D4F7 20 74 D5      t.     JSR  &D574
D4FA A0 00        ..      LDY  #&00
D4FC 20 AC D5      ..     JSR  &D5AC
D4FF A0 20        .       LDY  #&20
D501 A2 24        .$      LDX  #&24
D503 20 E6 CD      ..     JSR  &CDE6
```

```
D506 20 AA D4      ..      JSR  &D4AA
D509 A2 04         ..      LDX  #&04
D50B 20 92 D5      ..      JSR  &D592
D50E 8A            .       TXA
D50F D0 02         ..      BNE  &D513
D511 C6 DB         ..      DEC  &DB
D513 CA            .       DEX
D514 20 4B D5       K.     JSR  &D54B
D517 90 27         .'      BCC  &D540
D519 20 ED D3      ..      JSR  &D3ED
D51C B1 D6         ..      LDA  (&D6),Y
D51E 4D 5A 03      MZ.     EOR  &035A
D521 85 DA         ..      STA  &DA
D523 A5 DC         ..      LDA  &DC
D525 D0 ED         ..      BNE  &D514
D527 A5 DA         ..      LDA  &DA
D529 D0 12         ..      BNE  &D53D
D52B 38            8       SEC
D52C 8A            .       TXA
D52D 6D 61 03      ma.     ADC  &0361
D530 90 04         ..      BCC  &D536
D532 E6 DB         ..      INC  &DB
D534 10 07         ..      BPL  &D53D
D536 AA            .       TAX
D537 20 04 D1      ..      JSR  &D104
D53A 38            8       SEC
D53B B0 DC         ..      BCS  &D519
D53D 20 4B D5       K.     JSR  &D54B
D540 A0 04         ..      LDY  #&04
D542 20 AC D5      ..      JSR  &D5AC
D545 20 D9 D0      ..      JSR  &D0D9
D548 4C B8 D1      L..     JMP  &D1B8
D54B A5 D1         ..      LDA  &D1
D54D 48            H       PHA
D54E 18            .       CLC
D54F 90 0F         ..      BCC  &D560
D551 68            h       PLA
D552 E8            .       INX
D553 D0 04         ..      BNE  &D559
D555 E6 DB         ..      INC  &DB
D557 10 16         ..      BPL  &D56F
D559 46 D1         F.      LSR  &D1
D55B B0 12         ..      BCS  &D56F
D55D 05 D1         ..      ORA  &D1
D55F 48            H       PHA
D560 A5 D1         ..      LDA  &D1
D562 24 DA         $.      BIT  &DA
D564 08            .       PHP
D565 68            h       PLA
D566 45 DC         E.      EOR  &DC
D568 48            H       PHA
D569 28            (       PLP
D56A F0 E5         ..      BEQ  &D551
D56C 68            h       PLA
D56D 45 D1         E.      EOR  &D1
D56F 85 D1         ..      STA  &D1
D571 4C F0 D0      L..     JMP  &D0F0
D574 A9 00         ..      LDA  #&00
D576 18            .       CLC
D577 90 0A         ..      BCC  &D583
D579 E8            .       INX
D57A D0 04         ..      BNE  &D580
```

```
D57C E6 DB        ..      INC  &DB
D57E 10 EF        ..      BPL  &D56F
D580 0A           .       ASL  A
D581 B0 0B        ..      BCS  &D58E
D583 05 D1        ..      ORA  &D1
D585 24 DA        $.      BIT  &DA
D587 F0 F0        ..      BEQ  &D579
D589 45 D1        E.      EOR  &D1
D58B 4A           J       LSR  A
D58C 90 E1        ..      BCC  &D56F
D58E 6A           j       ROR  A
D58F 38           8       SEC
D590 B0 DD        ..      BCS  &D56F
D592 BD 00 03     ...     LDA  &0300,X
D595 38           8       SEC
D596 ED 20 03     . .     SBC  &0320
D599 A8           .       TAY
D59A BD 01 03     ...     LDA  &0301,X
D59D ED 21 03     .!.     SBC  &0321
D5A0 30 03        0.      BMI  &D5A5
D5A2 20 9B D4      ..     JSR  &D49B
D5A5 85 DB        ..      STA  &DB
D5A7 98           .       TYA
D5A8 AA           .       TAX
D5A9 05 DB        ..      ORA  &DB
D5AB 60           `       RTS
D5AC 84 DA        ..      STY  &DA
D5AE 8A           .       TXA
D5AF A8           .       TAY
D5B0 A5 DB        ..      LDA  &DB
D5B2 30 02        0.      BMI  &D5B6
D5B4 A9 00        ..      LDA  #&00
D5B6 A6 DA        ..      LDX  &DA
D5B8 D0 03        ..      BNE  &D5BD
D5BA 20 9B D4      ..     JSR  &D49B
D5BD 48           H       PHA
D5BE 18           .       CLC
D5BF 98           .       TYA
D5C0 7D 00 03     }..     ADC  &0300,X
D5C3 8D 20 03     . .     STA  &0320
D5C6 68           h       PLA
D5C7 7D 01 03     }..     ADC  &0301,X
D5CA 8D 21 03     .!.     STA  &0321
D5CD 60           `       RTS
D5CE A9 03        ..      LDA  #&03
D5D0 20 D5 D5      ..     JSR  &D5D5
D5D3 A9 07        ..      LDA  #&07
D5D5 48           H       PHA
D5D6 20 E2 CD      ..     JSR  &CDE2
D5D9 20 B8 D1      ..     JSR  &D1B8
D5DC A2 03        ..      LDX  #&03
D5DE 68           h       PLA
D5DF A8           .       TAY
D5E0 BD 10 03     ...     LDA  &0310,X
D5E3 91 F0        ..      STA  (&F0),Y
D5E5 88           .       DEY
D5E6 CA           .       DEX
D5E7 10 F7        ..      BPL  &D5E0
D5E9 60           `       RTS
D5EA A2 20        .       LDX  #&20
D5EC A0 3E        .>      LDY  #&3E
D5EE 20 7C D4      |.     JSR  &D47C
```

OSWORD &D  Read last two graphics.

```
D5F1 20 32 D6      2.     JSR  &D632
D5F4 A2 14         ..     LDX  #&14
D5F6 A0 24         .$     LDY  #&24
D5F8 20 36 D6      6.     JSR  &D636
D5FB 20 32 D6      2.     JSR  &D632
D5FE A2 20         .      LDX  #&20
D600 A0 2A         .*     LDY  #&2A
D602 20 11 D4      ..     JSR  &D411
D605 AD 2B 03      .+.    LDA  &032B
D608 8D 32 03      .2.    STA  &0332
D60B A2 28         .(     LDX  #&28
D60D 20 59 D4      Y.     JSR  &D459
D610 A0 2E         ..     LDY  #&2E
D612 20 DE D0      ..     JSR  &D0DE
D615 20 E2 CD      ..     JSR  &CDE2
D618 18            .      CLC
D619 20 58 D6      X.     JSR  &D658
D61C 20 E2 CD      ..     JSR  &CDE2
D61F A2 20         .      LDX  #&20
D621 20 E4 CD      ..     JSR  &CDE4
D624 38            8      SEC
D625 20 58 D6      X.     JSR  &D658
D628 A2 3E         .>     LDX  #&3E
D62A A0 20         .      LDY  #&20
D62C 20 7C D4      |.     JSR  &D47C
D62F 4C D9 D0      L..    JMP  &D0D9
D632 A2 20         .      LDX  #&20
D634 A0 14         ..     LDY  #&14
D636 BD 02 03      ...    LDA  &0302,X
D639 D9 02 03      ...    CMP  &0302,Y
D63C BD 03 03      ...    LDA  &0303,X
D63F F9 03 03      ...    SBC  &0303,Y
D642 30 13         0.     BMI  &D657
D644 4C E6 CD      L..    JMP  &CDE6
```

```
D647 AD 18 03      ...    LDA  &0318
D64A 38            8      SEC
D64B ED 08 03      ...    SBC  &0308
D64E AA            .      TAX
D64F AD 19 03      ...    LDA  &0319
D652 38            8      SEC
D653 ED 0B 03      ...    SBC  &030B
D656 A8            .      TAY
D657 60            `      RTS
```

*FX 134 Read text cursor position (POS+ VPOS).

```
D658 08            .      PHP
D659 A2 20         .      LDX  #&20
D65B A0 35         .5     LDY  #&35
D65D 20 11 D4      ..     JSR  &D411
D660 AD 36 03      .6.    LDA  &0336
D663 8D 3D 03      .=.    STA  &033D
D666 A2 33         .3     LDX  #&33
D668 20 59 D4      Y.     JSR  &D459
D66B A0 39         .9     LDY  #&39
D66D 20 DE D0      ..     JSR  &D0DE
D670 38            8      SEC
D671 AD 22 03      ."."   LDA  &0322
D674 ED 26 03      .&.    SBC  &0326
D677 8D 1B 03      ...    STA  &031B
D67A AD 23 03      .#.    LDA  &0323
D67D ED 27 03      .'.    SBC  &0327
D680 8D 1C 03      ...    STA  &031C
D683 0D 1B 03      ...    ORA  &031B
D686 F0 17         ..     BEQ  &D69F
```

```
D688 20 A2 D6      ..     JSR  &D6A2
D68B A2 33         .3     LDX  #&33
D68D 20 74 D7       t.    JSR  &D774
D690 A2 28         .(     LDX  #&28
D692 20 74 D7       t.    JSR  &D774
D695 EE 1B 03      ...    INC  &031B
D698 D0 EE         ..     BNE  &D688
D69A EE 1C 03      ...    INC  &031C
D69D D0 E9         ..     BNE  &D688
D69F 28            (      PLP
D6A0 90 B5         ..     BCC  &D657
D6A2 A2 39         .9     LDX  #&39
D6A4 A0 2E         ..     LDY  #&2E
D6A6 86 DE         ..     STX  &DE
D6A8 BD 00 03      ...    LDA  &0300,X
D6AB D9 00 03      ...    CMP  &0300,Y
D6AE BD 01 03      ...    LDA  &0301,X
D6B1 F9 01 03      ...    SBC  &0301,Y
D6B4 30 06         0.     BMI  &D6BC
D6B6 98            .      TYA
D6B7 A4 DE         ..     LDY  &DE
D6B9 AA            .      TAX
D6BA 86 DE         ..     STX  &DE
D6BC 84 DF         ..     STY  &DF
D6BE B9 00 03      ...    LDA  &0300,Y
D6C1 48            H      PHA
D6C2 B9 01 03      ...    LDA  &0301,Y
D6C5 48            H      PHA
D6C6 A6 DF         ..     LDX  &DF
D6C8 20 0F D1       ..    JSR  &D10F
D6CB F0 0D         ..     BEQ  &D6DA
D6CD C9 02         ..     CMP  #&02
D6CF D0 3D         .=     BNE  &D70E
D6D1 A2 04         ..     LDX  #&04
D6D3 A4 DF         ..     LDY  &DF
D6D5 20 82 D4       ..    JSR  &D482
D6D8 A6 DF         ..     LDX  &DF
D6DA 20 64 D8       d.    JSR  &D864
D6DD A6 DE         ..     LDX  &DE
D6DF 20 0F D1       ..    JSR  &D10F
D6E2 4A            J      LSR  A
D6E3 D0 29         .)     BNE  &D70E
D6E5 90 02         ..     BCC  &D6E9
D6E7 A2 00         ..     LDX  #&00
D6E9 A4 DF         ..     LDY  &DF
D6EB 38            8      SEC
D6EC B9 00 03      ...    LDA  &0300,Y
D6EF FD 00 03      ...    SBC  &0300,X
D6F2 85 DC         ..     STA  &DC
D6F4 B9 01 03      ...    LDA  &0301,Y
D6F7 FD 01 03      ...    SBC  &0301,X
D6FA 85 DD         ..     STA  &DD
D6FC A9 00         ..     LDA  #&00
D6FE 0A            .      ASL  A
D6FF 05 D1         ..     ORA  &D1
D701 A4 DC         ..     LDY  &DC
D703 D0 14         ..     BNE  &D719
D705 C6 DD         ..     DEC  &DD
D707 10 10         ..     BPL  &D719
D709 85 D1         ..     STA  &D1
D70B 20 F0 D0       ..    JSR  &D0F0
D70E A6 DF         ..     LDX  &DF
```

```
D710 68          h      PLA
D711 9D 01 03    ...    STA  &0301,X
D714 68          h      PLA
D715 9D 00 03    ...    STA  &0300,X
D718 60          `      RTS
D719 C6 DC       ..     DEC  &DC
D71B AA          .      TAX
D71C 10 E0       ..     BPL  &D6FE
D71E 85 D1       ..     STA  &D1
D720 20 F0 D0     ..    JSR  &D0F0
D723 A6 DC       ..     LDX  &DC
D725 E8          .      INX
D726 D0 02       ..     BNE  &D72A
D728 E6 DD       ..     INC  &DD
D72A 8A          .      TXA
D72B 48          H      PHA
D72C 46 DD       F.     LSR  &DD
D72E 6A          j      ROR  A
D72F AC 61 03    .a.    LDY  &0361
D732 C0 03       ..     CPY  #&03
D734 F0 05       ..     BEQ  &D73B
D736 90 06       ..     BCC  &D73E
D738 46 DD       F.     LSR  &DD
D73A 6A          j      ROR  A
D73B 46 DD       F.     LSR  &DD
D73D 4A          J      LSR  A
D73E AC 1A 03    ...    LDY  &031A
D741 AA          .      TAX
D742 F0 0F       ..     BEQ  &D753
D744 98          .      TYA
D745 38          8      SEC
D746 E9 08       ..     SBC  #&08
D748 A8          .      TAY
D749 B0 02       ..     BCS  &D74D
D74B C6 D7       ..     DEC  &D7
D74D 20 04 D1     ..    JSR  &D104
D750 CA          .      DEX
D751 D0 F1       ..     BNE  &D744
D753 68          h      PLA
D754 2D 61 03    -a.    AND  &0361
D757 F0 B5       ..     BEQ  &D70E
D759 AA          .      TAX
D75A A9 00       ..     LDA  #&00
D75C 0A          .      ASL  A
D75D 0D 63 03    .c.    ORA  &0363
D760 CA          .      DEX
D761 D0 F9       ..     BNE  &D75C
D763 85 D1       ..     STA  &D1
D765 98          .      TYA
D766 38          8      SEC
D767 E9 08       ..     SBC  #&08
D769 A8          .      TAY
D76A B0 02       ..     BCS  &D76E
D76C C6 D7       ..     DEC  &D7
D76E 20 F3 D0     ..    JSR  &D0F3
D771 4C 0E D7    L..    JMP  &D70E
D774 FE 08 03    ...    INC  &0308,X
D777 D0 03       ..     BNE  &D77C
D779 FE 09 03    ...    INC  &0309,X
D77C 38          8      SEC
D77D BD 00 03    ...    LDA  &0300,X
D780 FD 02 03    ...    SBC  &0302,X
```

```
D783 9D 00 03     ...     STA  &0300,X
D786 BD 01 03     ...     LDA  &0301,X
D789 FD 03 03     ...     SBC  &0303,X
D78C 9D 01 03     ...     STA  &0301,X
D78F 10 30        .0      BPL  &D7C1
D791 BD 0A 03     ...     LDA  &030A,X
D794 30 0B        0.      BMI  &D7A1
D796 FE 06 03     ...     INC  &0306,X
D799 D0 11        ..      BNE  &D7AC
D79B FE 07 03     ...     INC  &0307,X
D79E 4C AC D7     L..     JMP  &D7AC
D7A1 BD 06 03     ...     LDA  &0306,X
D7A4 D0 03        ..      BNE  &D7A9
D7A6 DE 07 03     ...     DEC  &0307,X
D7A9 DE 06 03     ...     DEC  &0306,X
D7AC 18           .       CLC
D7AD BD 00 03     ...     LDA  &0300,X
D7B0 7D 04 03     }..     ADC  &0304,X
D7B3 9D 00 03     ...     STA  &0300,X
D7B6 BD 01 03     ...     LDA  &0301,X
D7B9 7D 05 03     }..     ADC  &0305,X
D7BC 9D 01 03     ...     STA  &0301,X
D7BF 30 D0        0.      BMI  &D791
D7C1 60           `       RTS
D7C2 AC 60 03     .`.     LDY  &0360
D7C5 D0 15        ..      BNE  &D7DC
D7C7 B1 D8        ..      LDA  (&D8),Y
D7C9 A0 02        ..      LDY  #&02
D7CB D9 B7 C4     ...     CMP  &C4B7,Y
D7CE D0 04        ..      BNE  &D7D4
D7D0 B9 B6 C4     ...     LDA  &C4B6,Y
D7D3 88           .       DEY
D7D4 88           .       DEY
D7D5 10 F4        ..      BPL  &D7CB
D7D7 AC 55 03     .U.     LDY  &0355
D7DA AA           .       TAX
D7DB 60           `       RTS
D7DC 20 08 D8       .     JSR  &D808
D7DF A2 20        .       LDX  #&20
D7E1 8A           .       TXA
D7E2 48           H       PHA
D7E3 20 3E D0       >.    JSR  &D03E
D7E6 68           h       PLA
D7E7 AA           .       TAX
D7E8 A0 07        ..      LDY  #&07
D7EA B9 28 03     .(.     LDA  &0328,Y
D7ED D1 DE        ..      CMP  (&DE),Y
D7EF D0 08        ..      BNE  &D7F9
D7F1 88           .       DEY
D7F2 10 F6        ..      BPL  &D7EA
D7F4 8A           .       TXA
D7F5 E0 7F        ..      CPX  #&7F
D7F7 D0 DE        ..      BNE  &D7D7
D7F9 E8           .       INX
D7FA A5 DE        ..      LDA  &DE
D7FC 18           .       CLC
D7FD 69 08        i.      ADC  #&08
D7FF 85 DE        ..      STA  &DE
D801 D0 E5        ..      BNE  &D7E8
D803 8A           .       TXA
D804 D0 DB        ..      BNE  &D7E1
D806 F0 CF        ..      BEQ  &D7D7
```

*FX 135 Read Character at text cursor position.

Is it not Mode 7?

Teletext Conversion.

Copy character defn into &328 - &32F

Start with Space (32)

&DE, &DF = base address of char defn.

X = current char

next char number

Add 8 to char defn pointer

I'm sure this branch will always occur.

```
D808 A0 07        ..    LDY  #&07
D80A 84 DA        ..    STY  &DA
D80C A9 01        ..    LDA  #&01
D80E 85 DB        ..    STA  &DB
D810 AD 62 03     .b.   LDA  &0362
D813 85 DC        ..    STA  &DC
D815 B1 D8        ..    LDA  (&D8),Y
D817 4D 58 03     MX.   EOR  &0358
D81A 18           .     CLC
D81B 24 DC        $.    BIT  &DC
D81D F0 01        ..    BEQ  &D820
D81F 38           8     SEC
D820 26 DB        &.    ROL  &DB
D822 B0 0A        ..    BCS  &D82E
D824 46 DC        F.    LSR  &DC
D826 90 F3        ..    BCC  &D81B
D828 98           .     TYA
D829 69 07        i.    ADC  #&07
D82B A8           .     TAY
D82C 90 E2        ..    BCC  &D810
D82E A4 DA        ..    LDY  &DA
D830 A5 DB        ..    LDA  &DB
D832 99 28 03     .(.   STA  &0328,Y
D835 88           .     DEY
D836 10 D2        ..    BPL  &D80A
D838 60           `     RTS
D839 48           H     PHA
D83A AA           .     TAX
D83B 20 49 D1      I.   JSR  &D149
D83E 68           h     PLA
D83F AA           .     TAX
D840 20 5F D8      _.   JSR  &D85F
D843 D0 15        ..    BNE  &D85A
D845 B1 D6        ..    LDA  (&D6),Y
D847 0A           .     ASL  A
D848 26 DA        &.    ROL  &DA
D84A 06 D1        ..    ASL  &D1
D84C 08           .     PHP
D84D B0 02        ..    BCS  &D851
D84F 46 DA        F.    LSR  &DA
D851 28           (     PLP
D852 D0 F3        ..    BNE  &D847
D854 A5 DA        ..    LDA  &DA
D856 2D 60 03     -`.   AND  &0360
D859 60           `     RTS
D85A A9 FF        ..    LDA  #&FF
D85C 60           `     RTS
D85D A2 20        .     LDX  #&20
D85F 20 0F D1      ..   JSR  &D10F
D862 D0 F8        ..    BNE  &D85C
D864 BD 02 03     ...   LDA  &0302,X
D867 49 FF        I.    EOR  #&FF
D869 A8           .     TAY
D86A 29 07        ).    AND  #&07
D86C 8D 1A 03     ...   STA  &031A
D86F 98           .     TYA
D870 4A           J     LSR  A
D871 4A           J     LSR  A
D872 4A           J     LSR  A
D873 0A           .     ASL  A
D874 A8           .     TAY
D875 B1 E0        ..    LDA  (&E0),Y
```

Copy character down for character at cursor into &322 → &321

always taken

check for window violation - exit if true

Read internal co-ord of Y

invert → Y

&31A = Y AND 7 (origin TOP left is converted)

A = Y/8 then * 2 for look up table → Y

Multiplication table for current mode (High byte)

```
D877 85 DA       ..     STA  &DA      high byte of Multiple table.
D879 C8          .      INY
D87A B1 E0       ..     LDA  (&E0),Y  low of Mult. offset
D87C AC 56 03    .V.    LDY  &0356    memory map type.
D87F F0 03       ..     BEQ  &D884    branch if mode 7
D881 46 DA       F.     LSR  &DA      } divide by 2.
D883 6A          j      ROR  A        }
D884 6D 50 03    mP.    ADC  &0350    add low of screen ~~base~~ Start addr.
D887 85 D6       ..     STA  &D6      save
D889 A5 DA       ..     LDA  &DA      read high of Mult. table
D88B 6D 51 03    mQ.    ADC  &0351    add high of screen Start addr.
D88E 85 D7       ..     STA  &D7      save.
D890 BD 01 03    ...    LDA  &0301,X  read 'X' high (internal co-ord)
D893 85 DA       ..     STA  &DA      save
D895 BD 00 03    ...    LDA  &0300,X  read 'X' low (internal) & push.
D898 48          H      PHA
D899 2D 61 03    -a.    AND  &0361    } No of pixels(-1) per byte
D89C 6D 61 03    ma.    ADC  &0361    }
D89F A8          .      TAY           → Y
D8A0 B9 06 C4    ...    LDA  &C406,Y  Save pixel position in byte mask!
D8A3 85 D1       ..     STA  &D1
D8A5 68          h      PLA           recover 'X' low
D8A6 AC 61 03    .a.    LDY  &0361    No of pixels(-1) per byte.
D8A9 C0 03       ..     CPY  #&03     if Mode 1,5 branch
D8AB F0 05       ..     BEQ  &D8B2    if Mode 0,3,4,6
D8AD B0 06       ..     BCS  &D8B5
D8AF 0A          .      ASL  A        divide low of offset by 2
D8B0 26 DA       &.     ROL  &DA      " " " " " "
D8B2 0A          .      ASL  A
D8B3 26 DA       &.     ROL  &DA
D8B5 29 F8       ).     AND  #&F8
D8B7 18          .      CLC           low of Final address
D8B8 65 D6       e.     ADC  &D6
D8BA 85 D6       ..     STA  &D6
D8BC A5 DA       ..     LDA  &DA
D8BE 65 D7       e.     ADC  &D7      high byte of final address (wrap around checked for)
D8C0 10 04       ..     BPL  &D8C6
D8C2 38          8      SEC
D8C3 ED 54 03    .T.    SBC  &0354
D8C6 85 D7       ..     STA  &D7
D8C8 AC 1A 03    ...    LDY  &031A    8byte
D8CB A9 00       ..     LDA  #&00     A=0, Y=column offset on exit.
D8CD 60          `      RTS
D8CE 48          H      PHA
D8CF A9 A0       ..     LDA  #&A0
D8D1 AE 6A 02    .j.    LDX  &026A
D8D4 D0 40       .@     BNE  &D916
D8D6 24 D0       $.     BIT  &D0
D8D8 D0 3C       .<     BNE  &D916
D8DA 70 19       p.     BVS  &D8F5
D8DC AD 5F 03    ._.    LDA  &035F
D8DF 29 9F       ).     AND  #&9F
D8E1 09 40       .@     ORA  #&40
D8E3 20 54 C9     T.    JSR  &C954
D8E6 A2 18       ..     LDX  #&18
D8E8 A0 64       .d     LDY  #&64
D8EA 20 82 D4     ..    JSR  &D482
D8ED 20 7A CD     z.    JSR  &CD7A
D8F0 A9 02       ..     LDA  #&02
D8F2 20 9D C5     ..    JSR  &C59D
D8F5 A9 BF       ..     LDA  #&BF
D8F7 20 A8 C5     ..    JSR  &C5A8
```

```
D8FA 68          h       PLA
D8FB 29 7F       ).      AND  #&7F
D8FD 20 C0 C4     ..     JSR  &C4C0
D900 A9 40       .@      LDA  #&40
D902 4C 9D C5    L..     JMP  &C59D
D905 A9 20       .       LDA  #&20
D907 24 D0       $.      BIT  &D0
D909 50 C0       P.      BVC  &D8CB
D90B D0 BE       ..      BNE  &D8CB
D90D 20 C2 D7     ..     JSR  &D7C2
D910 F0 05       ..      BEQ  &D917
D912 48          H       PHA
D913 20 64 C6     d.     JSR  &C664
D916 68          h       PLA
D917 60          `       RTS
```

Process VDU 13 (CR)

```
D918 A9 BD       ..      LDA  #&BD
D91A 20 A8 C5     ..     JSR  &C5A8
D91D 20 51 C9     Q.     JSR  &C951
D920 A9 0D       ..      LDA  #&0D
D922 60          `       RTS
```

- clear bits 6 & 1. (cursors together) see &C4D5
- poke cursor start reg. & scrolling enabled (?)
A: CR (13)

```
D923 AE 55 03    .U.     LDX  &0355
D926 8A          .       TXA
D927 29 07       ).      AND  #&07
D929 A8          .       TAY
D92A BE 40 C4    .@.     LDX  &C440,Y
D92D BD 5E C4    .^.     LDA  &C45E,X
D930 A2 00       ..      LDX  #&00
D932 2C 8E 02    ,..     BIT  &028E
D935 30 07       0.      BMI  &D93E
D937 29 3F       )?      AND  #&3F
D939 C0 04       ..      CPY  #&04
D93B B0 01       ..      BCS  &D93E
D93D 8A          .       TXA
D93E A8          .       TAY
D93F 60          `       RTS
```

*FX 132 — Read Bottom of display RAM address (HIMEM).

*FX133 — As above, but for any Mode.

```
D940 10 E3       ..      BPL  &D925
D942 54          T       ???
D943 DC          .       ???
D944 93          .       ???
D945 DC          .       ???
D946 89          .       ???
D947 DE 89 DF    ...     DEC  &DF89,X
D94A 72          r       ???
D94B E7          .       ???
D94C EB          .       ???
D94D E7          .       ???
D94E A4 E0       ..      LDY  &E0
D950 C5 DE       ..      CMP  &DE
D952 7D F2 8E    }..     ADC  &8EF2,X
D955 F1 C9       ..      SBC  (&C9),Y
D957 F4          .       ???
D958 29 F5       ).      AND  #&F5
D95A A6 FF       ..      LDX  &FF
D95C CA          .       DEX
D95D F3          .       ???
D95E B1 F1       ..      LDA  (&F1),Y
D960 A6 FF       ..      LDX  &FF
D962 A6 FF       ..      LDX  &FF
D964 A6 FF       ..      LDX  &FF
D966 A6 FF       ..      LDX  &FF
D968 02          .       ???
D969 EF          .       ???
```

Vector table. (&36 bytes)

locations

Data copied into &212+ see &F159

```
D96A B3          .      ???
D96B E4 64       .d     CPX  &64
D96D E4 D1       ..     CPX  &D1
D96F E1 A6       ..     SBC  (&A6,X)
D971 FF          .      ???
D972 A6 FF       ..     LDX  &FF
D974 A6 FF       ..     LDX  &FF
D976 90 01       ..     BCC  &D979
D978 9F          .      ???
D979 0D A1 02    ...    ORA  &02A1
D97C 2B          +      ???
D97D F0 00       ..     BEQ  &D97F
D97F 03          .      ???
D980 00          .      BRK
D981 00          .      BRK
D982 FF          .      ???
 .       BRK
D985 01 00       ..     ORA  (&00,X)
D987 00          .      BRK
D988 00          .      BRK
D989 00          .      BRK
D98A 00          .      BRK
D98B FF          .      ???
D98C 04          .      ???
D98D 04          .      ???
D98E 00          .      BRK
D98F FF          .      ???
D990 56 19       V.     LSR  &19,X
D992 19 19 32    ..2    ORA  &3219,Y
D995 08          .      PHP
D996 00          .      BRK
D997 00          .      BRK
D998 00          .      BRK
D999 00          .      BRK
D99A 20 09 00     ..    JSR  &0009
D99D 00          .      BRK
D99E 00          .      BRK
D99F 00          .      BRK
D9A0 00          .      BRK
D9A1 50 00       P.     BVC  &D9A3
D9A3 03          .      ???
D9A4 90 64       .d     BCC  &DA0A
D9A6 06 81       ..     ASL  &81
D9A8 00          .      BRK
D9A9 00          .      BRK
D9AA 00          .      BRK
D9AB 09 1B       ..     ORA  #&1B
D9AD 01 D0       ..     ORA  (&D0,X)
D9AF E0 F0       ..     CPX  #&F0
D9B1 01 80       ..     ORA  (&80,X)
D9B3 90 00       ..     BCC  &D9B5
D9B5 00          .      BRK
D9B6 00          .      BRK
D9B7 FF          .      ???
D9B8 FF          .      ???
D9B9 FF          .      ???
D9BA 00          .      BRK
D9BB 00          .      BRK
D9BC 00          .      BRK
D9BD 00          .      BRK
D9BE 00          .      BRK
D9BF 00          .      BRK
```

```
D9C0 00          .      BRK
D9C1 00          .      BRK
D9C2 64          d      ???
D9C3 05 FF       ..     ORA   &FF
D9C5 01 0A       ..     ORA   (&0A,X)
D9C7 00          .      BRK
D9C8 00          .      BRK
D9C9 00          .      BRK
D9CA 00          .      BRK
D9CB 00          .      BRK
D9CC FF          .      ???
D9CD A9 40       .@     LDA   #&40
D9CF 8D 00 0D    ...    STA   &0D00
D9D2 78          x      SEI
D9D3 D8          .      CLD
D9D4 A2 FF       ..     LDX   #&FF
D9D6 9A          .      TXS
D9D7 AD 4E FE    .N.    LDA   &FE4E
D9DA 0A          .      ASL   A
D9DB 48          H      PHA
D9DC F0 09       ..     BEQ   &D9E7
D9DE AD 58 02    .X.    LDA   &0258
D9E1 4A          J      LSR   A
D9E2 C9 01       ..     CMP   #&01
D9E4 D0 1D       ..     BNE   &DA03
D9E6 4A          J      LSR   A
D9E7 A2 04       ..     LDX   #&04
D9E9 86 01       ..     STX   &01
D9EB 85 00       ..     STA   &00
D9ED A8          .      TAY
D9EE 91 00       ..     STA   (&00),Y
D9F0 C5 01       ..     CMP   &01
D9F2 F0 09       ..     BEQ   &D9FD
D9F4 C8          .      INY
D9F5 D0 F7       ..     BNE   &D9EE
D9F7 C8          .      INY
D9F8 E8          .      INX
D9F9 E6 01       ..     INC   &01
D9FB 10 F1       ..     BPL   &D9EE
D9FD 8E 8E 02    ...    STX   &028E
DA00 8E 84 02    ...    STX   &0284
DA03 A2 0F       ..     LDX   #&0F
DA05 8E 42 FE    .B.    STX   &FE42
DA08 CA          .      DEX
DA09 8E 40 FE    .@.    STX   &FE40
DA0C E0 09       ..     CPX   #&09
DA0E B0 F8       ..     BCS   &DA08
DA10 E8          .      INX
DA11 8A          .      TXA
DA12 20 2A F0     *.    JSR   &F02A
DA15 E0 80       ..     CPX   #&80
DA17 66 FC       f.     ROR   &FC
DA19 AA          .      TAX
DA1A CA          .      DEX
DA1B D0 F4       ..     BNE   &DA11
DA1D 8E 8D 02    ...    STX   &028D
DA20 26 FC       &.     ROL   &FC
DA22 20 EB EE     ..    JSR   &EEEB
DA25 6A          j      ROR   A
DA26 A2 9C       ..     LDX   #&9C
DA28 A0 8D       ..     LDY   #&8D
DA2A 68          h      PLA
```

NMI RTI

RESET.

} Disable interrupts + clear Decimal mode.

} SP = &FF

Interrupt enable register for System VIA.

} IF IER*2 = 0 clear memory

} Bit 1 set? (Not if bits 2-7 set)

Clear memory &4000 - &7FFF

} = &80 soft key consistency. RAM available.

} Port A, I/O, System VIA = %1111. (Data direction

} Port B, Output, System VIA = %11110. %10001

} Acc = 9, X = 9

} Get key press (-ve). Bit 7 of &FC set if

} repeated for X, 9-1 Indusiv

Break type (soft, power up, hard) = 0.
Used as general purpose.
Set Caps lock, Shift lock LED's as required.

Acc = IFR (System VIA) see &D9DB

```
DA2B F0 09       ..      BEQ  &DA36
DA2D A0 7E       .~      LDY  #&7E
DA2F 90 11       ..      BCC  &DA42
DA31 A0 87       ..      LDY  #&87
DA33 EE 8D 02    ...     INC  &028D
DA36 EE 8D 02    ...     INC  &028D
DA39 A5 FC       ..      LDA  &FC
DA3B 49 FF       I.      EOR  #&FF
DA3D 8D 8F 02    ...     STA  &028F
DA40 A2 90       ..      LDX  #&90
DA42 A9 00       ..      LDA  #&00
DA44 E0 CE       ..      CPX  #&CE
DA46 90 02       ..      BCC  &DA4A
DA48 A9 FF       ..      LDA  #&FF
DA4A 9D 00 02    ...     STA  &0200,X
DA4D E8          .       INX
DA4E D0 F4       ..      BNE  &DA44
DA50 8D 63 FE    .c.     STA  &FE63
DA53 8A          .       TXA
DA54 A2 E2       ..      LDX  #&E2
DA56 95 00       ..      STA  &00,X
DA58 E8          .       INX
DA59 D0 FB       ..      BNE  &DA56
DA5B B9 3F D9    .?.     LDA  &D93F,Y
DA5E 99 FF 01    ...     STA  &01FF,Y
DA61 88          .       DEY
DA62 D0 F7       ..      BNE  &DA5B
DA64 A9 62       .b      LDA  #&62
DA66 85 ED       ..      STA  &ED
DA68 20 0A FB     ..     JSR  &FB0A
DA6B A9 7F       ..      LDA  #&7F
DA6D E8          .       INX
DA6E 9D 4D FE    .M.     STA  &FE4D,X
DA71 9D 6D FE    .m.     STA  &FE6D,X
DA74 CA          .       DEX
DA75 10 F7       ..      BPL  &DA6E
DA77 58          X       CLI
DA78 78          x       SEI
DA79 24 FC       $.      BIT  &FC
DA7B 50 03       P.      BVC  &DA80
DA7D 20 55 F0     U.     JSR  &F055
DA80 A2 F2       ..      LDX  #&F2
DA82 8E 4E FE    .N.     STX  &FE4E
DA85 A2 04       ..      LDX  #&04
DA87 8E 4C FE    .L.     STX  &FE4C
DA8A A9 60       .`      LDA  #&60
DA8C 8D 4B FE    .K.     STA  &FE4B
DA8F A9 0E       ..      LDA  #&0E
DA91 8D 46 FE    .F.     STA  &FE46
DA94 8D 6C FE    .l.     STA  &FE6C
DA97 8D C0 FE    ...     STA  &FEC0
DA9A CD 6C FE    .l.     CMP  &FE6C
DA9D F0 03       ..      BEQ  &DAA2
DA9F EE 77 02    .w.     INC  &0277
DAA2 A9 27       .'      LDA  #&27
DAA4 8D 47 FE    .G.     STA  &FE47
DAA7 8D 45 FE    .E.     STA  &FE45
DAAA 20 60 EC     `.     JSR  &EC60
DAAD AD 82 02    ...     LDA  &0282
DAB0 29 7F       ).      AND  #&7F
DAB2 20 A7 E6     .      JSR  &E6A7
DAB5 AE 84 02    ...     LDX  &0284
```

50

If IER WAS ø then Branch (ON POWER UP ONLY)

If 'CTRL' Not pressed then branch (Sft break)

INCREMENT hard/soft 'BREAK'. &28D=1=Power up &28D=2-hard &28D=ø-Soft

'Not' Bits ø-7 of key pressed.

Start up options.

Clear System variables & vectors.

DATA Direction register 'A'

A=ø

&E2-&FF=ø

Copy vectors from ROM table (&D940+)

Internal key Number of the First key pressed.

Reset 6850

X for 1 & ø Incl. IFR & IER = &7F for USER & SYSTEM VIA.

Program in 1MHz bus to allow external Disc units etc

JMP(&FDFE) ???

System VIA (IER) = &F2

System VIA (Shift register) = 4

System VIA (Aux control register) = &60.

T1 low order latch = &E (System VIA)
Shift register = &E (USER VIA)
ADC Status register = &E
Shift register (USER VIA).

(IRQ bit mask for user 6522).

T1 high order latch = &27
T1 " " Counter = &27

copy of serial processor ULA register.
get bits ø-6 only.
write to serial processor ULA register.
read soft key consistency Flag.

```
DAB8 F0 03      ..      BEQ  &DABD      branch if in consistent state
DABA 20 C8 E9    ..     JSR  &E9C8      reset soft keys.
DABD 20 16 DC   ..>     JSR  &DC16      write ROM socket No to &F4, &FE30
DAC0 A2 03      ..      LDX  #&03                              (X reg)
DAC2 AC 07 80   ..^     LDY  &8007
DAC5 B9 00 80   ...     LDA  &8000,Y    } check for '(C)' message
DAC8 DD 0C DF   ...     CMP  &DF0C,X    }
DACB D0 2E      ..      BNE  &DAFB      Can't find '(C)'
DACD C8         .       INY
DACE CA         .       DEX
DACF 10 F4      ..      BPL  &DAC5
DAD1 A6 F4      ..      LDX  &F4        VALID ROM
DAD3 A4 F4      ..      LDY  &F4
DAD5 C8         .       INY
DAD6 C0 10      ..      CPY  #&10       } If socket 15 or > then check ID. (why skip
DAD8 B0 25      .%      BCS  &DAFF      }                                  next bit?
DADA 98         .       TYA             A= Socket No. + 1
DADB 49 FF      I.      EOR  #&FF       } LOW=(Socket No. + 1) EOR#&FF
DADD 85 FA      ..      STA  &FA
DADF A9 7F      ..^     LDA  #&7F       } High= &7F
DAE1 85 FB      ..      STA  &FB
DAE3 8C 30 FE   .0.     STY  &FE30      Select ROM (+1)
DAE6 B1 FA      ..      LDA  (&FA),Y    read byte
DAE8 8E 30 FE   .0.     STX  &FE30      Select ROM (+0)
DAEB D1 FA      ..^     CMP  (&FA),Y    Cmp code from N &(N+1) socket
DAED D0 E6      ..      BNE  &DAD5      if different then branch
DAEF E6 FA      ..      INC  &FA        Increment Low
DAF1 D0 F0      ..      BNE  &DAE3        " high if necessary.
DAF3 E6 FB      ..      INC  &FB
DAF5 A5 FB      ..      LDA  &FB
DAF7 C9 84      ..^     CMP  #&84
DAF9 90 E8      ..      BCC  &DAE3
DAFB A6 F4      ..      LDX  &F4
DAFD 10 0D      ..      BPL  &DB0C      — Always taken(?)
DAFF AD 06 80   ..^     LDA  &8006      } read ROM ID and save in ROM ID Table
DB02 9D A1 02   ...     STA  &02A1,X    }
DB05 29 8F      ).      AND  #&8F       } If bit 6 or 5 is set, then it's BASIC.
DB07 D0 03      ..      BNE  &DB0C      }
DB09 8E 4B 02   .K.     STX  &024B      Socket No. containing BASIC
DB0C E8         .       INX             Next ROM
DB0D E0 10      ..      CPX  #&10       } All 16 ROMS DONE? Branch if not.
DB0F 90 AC      ..      BCC  &DABD      }
DB11 2C 40 FE   ,@.     BIT  &FE40      } read REG Ø (Input reg B) System VIA.
DB14 30 11      0.      BMI  &DB27      }
DB16 CE 7B 02   .{.     DEC  &027B      -1 from speech processor present (to &FF
DB19 A0 FF      ..      LDY  #&FF       }                                 From Ø)
DB1B 20 7F EE    ..     JSR  &EE7F      } write to speech processor.
DB1E CA         .       DEX
DB1F D0 F8      ..      BNE  &DB19
DB21 8E 48 FE   .H.     STX  &FE48      timer 2 low counter =Ø System VIA.
DB24 8E 49 FE   .I.     STX  &FE49        "   "  high   "  =Ø
DB27 AD 8F 02   ...     LDA  &028F      read start up OPTIONS
DB2A 20 00 C3    ..     JSR  &C300      change to MODE 7
DB2D A0 CA      ..      LDY  #&CA
DB2F 20 F1 E4    ..     JSR  &E4F1      Insert character into input buffer
DB32 20 D9 EA    ..*    JSR  &EAD9      Check for BREAK intercept (C=Ø
DB35 20 40 F1    @.     JSR  &F140      Reset *OPT's, ROM service call (?)
DB38 A9 81      ..      LDA  #&81
DB3A 8D E0 FE   ...     STA  &FEE0      } Tube (test if present).
DB3D AD E0 FE   ...     LDA  &FEE0      }
DB40 6A         j       ROR  A          set carry according to bit Ø.
DB41 90 0A      ..      BCC  &DB4D      Branch if No 'Tube'.
```

bit Ø = 1 = 'Tube'
bit Ø = Ø = No 'Tube'

```
DB43 A2 FF      ..      LDX  #&FF   - service call &FF (TUBE present) Main Init.
DB45 20 68 F1    h.     JSR  &F168  Inform Paged ROMs of tube presence (service call
DB48 D0 03      ..      BNE  &DB4D  If Not accepted then branch & ignore 2P.
DB4A CE 7A 02   .z.     DEC  &027A  set &27A to &FF, indicating Tube present.
DB4D A0 0E      ..      LDY  #&0E   Start of RAM which can be claimed.
DB4F A2 01      ..      LDX  #&01   } issue service call to Paged ROMS. (call 1)
DB51 20 68 F1    h.     JSR  &F168  }                    Claim workspace, (Absolute
DB54 A2 02      ..      LDX  #&02   } service call 2 ( claim workspace, Private).
DB56 20 68 F1    h.     JSR  &F168  }
DB59 8C 43 02   .C.     STY  &0243  } Primary OSHWM (For imploded Font)
DB5C 8C 44 02   .D.     STY  &0244  }          ( " ).
DB5F A2 FE      ..      LDX  #&FE
DB61 AC 7A 02   .z.     LDY  &027A  Tube presence Flage } Tube post
DB64 20 68 F1    h.     JSR  &F168                       } Initialisation call.
DB67 2D 67 02   -g.     AND  &0267  Start up message suppression.
DB6A 10 1B      ..      BPL  &DB87  (branch & ignore 0.5 Start up message)
DB6C A0 02      ..      LDY  #&02
DB6E 20 A9 DE    ..     JSR  &DEA9  Print 'BBC Computer'
DB71 AD 8D 02   ...     LDA  &028D  read hard/soft break.
DB74 F0 0C      ..      BEQ  &DB82  branch if soft.
DB76 A0 16      ..      LDY  #&16
DB78 2C 8E 02   ,..     BIT  &028E  Available RAM
DB7B 30 02      0.      BMI  &DB7F  if 32K (Y=&16)
DB7D A0 11      ..      LDY  #&11   Y=&11 if 16K
DB7F 20 A9 DE    ..     JSR  &DEA9  Print '16K' or '32K'+Beep
DB82 A0 1B      ..      LDY  #&1B   } Print 2CR + Cursor Movement
DB84 20 A9 DE    ..     JSR  &DEA9  }
DB87 38         8       SEC                                   NB C=1 (2nd !)
DB88 20 D9 EA    ..  *  JSR  &EAD9  check For 'Break' intercept code
DB8B 20 D9 E9    ..     JSR  &E9D9  reflect keyboard status in LED's.
DB8E 08         .       PHP
DB8F 68         h       PLA
DB90 4A         J       LSR  A      } Divide by 16 ( get LSN)
DB91 4A         J       LSR  A      }
DB92 4A         J       LSR  A      }
DB93 4A         J       LSR  A      }
DB94 4D 8F 02   M..     EOR  &028F  Start up options.
DB97 29 08      ).      AND  #&08   test bit 3
DB99 A8         .       TAY         Y=0 or 8
DB9A A2 03      ..      LDX  #&03   X=3 } Auto boot service call
DB9C 20 68 F1    h.     JSR  &F168  }
DB9F F0 1D      ..      BEQ  &DBBE  if Accepted then branch. (by ROM eg. DFS)
DBA1 98         .       TYA
DBA2 D0 14      ..      BNE  &DBB8
DBA4 A9 8D      ..      LDA  #&8D   } *ROM call.   ROM Filing system
DBA6 20 35 F1    5.     JSR  &F135  }              boot option.
DBA9 A2 D2      ..      LDX  #&D2   } Point to '/!BOOT'
DBAB A0 EA      ..      LDY  #&EA   }
DBAD CE 67 02   .g.     DEC  &0267  Start up message Suppresion
DBB0 20 F7 FF    ..     JSR  OSCLI  Pass to command line interpreter
DBB3 EE 67 02   .g.     INC  &0267  Start up message suppresion.
DBB6 D0 06      ..      BNE  &DBBE
DBB8 A9 00      ..      LDA  #&00   } A, X=0
DBBA AA         .       TAX         }
DBBB 20 37 F1    7.     JSR  &F137  *TAPE call
DBBE AD 8D 02   ...     LDA  &028D  hard/soft break.
DBC1 D0 05      ..      BNE  &DBC8  branch if power up or hard
DBC3 AE 8C 02   ...     LDX  &028C  Current language ROM socket
DBC6 10 1E      ..      BPL  &DBE6  Always take branch! (?).
DBC8 A2 0F      ..      LDX  #&0F
DBCA BD A1 02   ...     LDA  &02A1,X } read ROM ID byte From ID table.
DBCD 2A         *       ROL  A      Acc *2. loose bit 7 into carry
```

```
DBCE 30 16        0.    BMI  &DBE6
DBD0 CA           .     DEX
DBD1 10 F7        ..    BPL  &DBCA
DBD3 A9 00        ..    LDA  #&00
DBD5 2C 7A 02     ,z.   BIT  &027A
DBD8 30 2E        0.    BMI  &DC08
DBDA 00           .     BRK
DBDB F9 4C 61     .La   SBC  &614C,Y
DBDE 6E 67 75     ngu   ROR  &7567
DBE1 61 67        ag    ADC  (&67,X)
DBE3 65 3F        e?    ADC  &3F
DBE5 00           .     BRK
DBE6 18           .     CLC
DBE7 08           .     PHP
DBE8 8E 8C 02     ...   STX  &028C
DBEB 20 16 DC      ..   JSR  &DC16
DBEE A9 80        ..    LDA  #&80
DBF0 A0 08        ..    LDY  #&08
DBF2 20 AB DE      ..   JSR  &DEAB
DBF5 84 FD        ..    STY  &FD
DBF7 20 E7 FF      ..   JSR  OSNEWL
DBFA 20 E7 FF      ..   JSR  OSNEWL
DBFD 28           (     PLP
DBFE A9 01        ..    LDA  #&01
DC00 2C 7A 02     ,z.   BIT  &027A
DC03 30 03        0.    BMI  &DC08
DC05 4C 00 80     L..   JMP  &8000
DC08 4C 00 04     L..   JMP  &0400
DC0B A6 F4        ..    LDX  &F4
DC0D 84 F4        ..    STY  &F4
DC0F 8C 30 FE     .0.   STY  &FE30
DC12 A0 00        ..    LDY  #&00
DC14 B1 F6        ..    LDA  (&F6),Y
DC16 86 F4        ..    STX  &F4
DC18 8E 30 FE     .0.   STX  &FE30
DC1B 60           `     RTS
DC1C 85 FC        ..    STA  &FC
DC1E 68           h     PLA
DC1F 48           H     PHA
DC20 29 10        ).    AND  #&10
DC22 D0 03        ..    BNE  &DC27
DC24 6C 04 02     l..   JMP  (&0204)
DC27 8A           .     TXA
DC28 48           H     PHA
DC29 BA           .     TSX
DC2A BD 03 01     ...   LDA  &0103,X
DC2D D8           .     CLD
DC2E 38           8     SEC
DC2F E9 01        ..    SBC  #&01
DC31 85 FD        ..    STA  &FD
DC33 BD 04 01     ...   LDA  &0104,X
DC36 E9 00        ..    SBC  #&00
DC38 85 FE        ..    STA  &FE
DC3A A5 F4        ..    LDA  &F4
DC3C 8D 4A 02     .J.   STA  &024A
DC3F 86 F0        ..    STX  &F0
DC41 A2 06        ..    LDX  #&06
DC43 20 68 F1      h.   JSR  &F168
DC46 AE 8C 02     ...   LDX  &028C
DC49 20 16 DC      ..   JSR  &DC16
DC4C 68           h     PLA
DC4D AA           .     TAX
```

Branch if bit 6 (before shift) is set.

Tube presence Flag.
If Tube there then branch.

'Language?' } No language ROM Found (bit6=1 if language)

Store Status. *FX142 Enter language ROM
current language ROM number.
select ROM & make 0.5 copy of ROM No.

} Print 'ROM' title eg 'BASIC'

&FD = low byte from Y reg.

} 2 CR's

recover status.

Tube present, branch if present.

enter language ROM with (A=1.)

enter TUBE code at &400. (A=0).

OSRDRM

read byte from ROM into Acc.

select ROM + make 0.5 copy.

(Main IRQ) (Interrupt routine BRK etc)

save Acc

} get & put status register

From &FFFE } low
&FFFF } high

check BRK bit.

branch if IRQ is from a BRK

} treat as a hardware interrupt.

} Save 'X'

get current S. pointer.

get byte off hardware stack

decimal

Subtract 1 from low byte

high

get byte off stack

← sub 1 & carry clear.

high of BRK address

read current paged ROM while BRK occ

save current ROM in &24A

save stack pointer

service call = 6 (BRK occurred)

issue paged ROM service call

read current language ROM socket num

change to language ROM

} remove 'X' from the stack.

```
DC4E A5 FC        ..      LDA   &FC
DC50 58           X       CLI
DC51 6C 02 02     l..     JMP   (&0202)
DC54 A0 00        ..      LDY   #&00
DC56 20 B1 DE      ..     JSR   &DEB1
DC59 AD 67 02     .g.     LDA   &0267
DC5C 6A           j       ROR   A
DC5D B0 FE        ..      BCS   &DC5D
DC5F 20 E7 FF      ..     JSR   OSNEWL
DC62 20 E7 FF      ..     JSR   OSNEWL
DC65 4C B8 DB     L..     JMP   &DBB8
DC68 38           8       SEC
DC69 6E 4F 02     nO.     ROR   &024F
DC6C 2C 50 02     ,P.     BIT   &0250
DC6F 10 07        ..      BPL   &DC78
DC71 20 41 E7      A.     JSR   &E741
DC74 A2 00        ..      LDX   #&00
DC76 B0 02        ..      BCS   &DC7A
DC78 A2 40        .@      LDX   #&40
DC7A 4C 7A E1     Lz.     JMP   &E17A
DC7D AC 09 FE     ...     LDY   &FE09
DC80 29 3A        ):      AND   #&3A
DC82 D0 34        .4      BNE   &DCB8
DC84 AE 5C 02     .\.     LDX   &025C
DC87 D0 09        ..      BNE   &DC92
DC89 E8           .       INX
DC8A 20 F3 E4      ..     JSR   &E4F3
DC8D 20 41 E7      A.     JSR   &E741
DC90 90 E6        ..      BCC   &DC78
DC92 60           `       RTS
DC93 D8           .       CLD
DC94 A5 FC        ..      LDA   &FC
DC96 48           H       PHA
DC97 8A           .       TXA
DC98 48           H       PHA
DC99 98           .       TYA
DC9A 48           H       PHA
DC9B A9 DE        ..      LDA   #&DE
DC9D 48           H       PHA
DC9E A9 81        ..      LDA   #&81
DCA0 48           H       PHA
DCA1 B8           .       CLV
DCA2 AD 08 FE     ...     LDA   &FE08
DCA5 70 02        p.      BVS   &DCA9
DCA7 10 5D        .]      BPL   &DD06
DCA9 A6 EA        ..      LDX   &EA
DCAB CA           .       DEX
DCAC 30 30        00      BMI   &DCDE
DCAE 70 2D        p-      BVS   &DCDD
DCB0 4C 88 F5     L..     JMP   &F588
DCB3 AC 09 FE     ...     LDY   &FE09
DCB6 2A           *       ROL   A
DCB7 0A           .       ASL   A
DCB8 AA           .       TAX
DCB9 98           .       TYA
DCBA A0 07        ..      LDY   #&07
DCBC 4C 94 E4     L..     JMP   &E494
DCBF A2 02        ..      LDX   #&02
DCC1 20 60 E4      `.     JSR   &E460
DCC4 90 10        ..      BCC   &DCD6
DCC6 AD 85 02     ...     LDA   &0285
DCC9 C9 02        ..      CMP   #&02
```

restore Accumulator
restore interrupts.
JUMP (BRK vector) - into Basic ROM
Current language ROM.

Two CR's

Clear decimal mode IRQ1 (From vector at &204 & 205.)
read Acc From &FC
Save Acc
Save X
Save Y
} on Stack

Put &DE81 on Stack - this restores registers & exits from interrupt routine through address

V=0

read control register from 6850
Bit 6 set? branch if 1 (Parity error bit)
Bit 7 set? branch if 0. (6850 IRQ bit) IRQ if 2
RS423 timeout counter

if result -ve, branch always taken - apparently (returns

data register (RX) - 6850

*2

X*2.

Y→A

Generate event 7 (RS423 error)

get character from buffer (2 - RS423 only)

branch if buffer contained a character

- printer destination flag.

```
DCCB D0 9B       ..      BNE  &DC68
DCCD E8          .       INX
DCCE 20 60 E4     `.     JSR  &E460     remove character from buffer (X=buffer
DCD1 6E D2 02    n..     ROR  &02D2     buffer busy flag clear (for Printer buffer
DCD4 30 92       0.      BMI  &DC68
DCD6 8D 09 FE    ...     STA  &FE09     TX data register. - 6850.
DCD9 A9 E7       ..      LDA  #&E7
DCDB 85 EA       ..      STA  &EA       - General W/space
DCDD 60          `       RTS            exit.
DCDE 2D 78 02    -x.     AND  &0278     6850 IRQ mask (RS423).
DCE1 4A          J       LSR  A         /2
DCE2 90 07       ..      BCC  &DCEB     branch if bit 0 was 0. (& Receive data regen
DCE4 70 05       p.      BVS  &DCEB     always taken - apparently.
DCE6 AC 50 02    .P.     LDY  &0250     RS423 control flag
DCE9 30 92       0.      BMI  &DC7D     branch if bit 7 is set
DCEB 4A          J       LSR  A         /2
DCEC 6A          j       ROR  A         /2
DCED B0 C4       ..      BCS  &DCB3
DCEF 30 CE       0.      BMI  &DCBF
DCF1 70 EA       p.      BVS  &DCDD     exit
DCF3 A2 05       ..      LDX  #&05      X=5. - service request = unrecognised interrupt.
DCF5 20 68 F1     h.     JSR  &F168     issue paged ROM service request.
DCF8 F0 E3       ..      BEQ  &DCDD     - branch if a ROM has accepted request. (RTS)
DCFA 68          h       PLA            } &DE81 off stack, see &DC9B+
DCFB 68          h       PLA            }
DCFC 68          h       PLA            } restore Y
DCFD A8          .       TAY            }
DCFE 68          h       PLA            } restore X
DCFF AA          .       TAX            }
DD00 68          h       PLA            restore A
DD01 85 FC       ..      STA  &FC       Save A
DD03 6C 06 02    1..     JMP  (&0206)   Jump through &206-&207 IRQ2V.
DD06 AD 4D FE    .M.     LDA  &FE4D     read interrupt flag register of System VIA
DD09 10 3C       .<      BPL  &DD47     branch if interrupt not from VIA.
DD0B 2D 79 02    -y.     AND  &0279     AND with 6522 System VIA mask
DD0E 2D 4E FE    -N.     AND  &FE4E     AND with interrupts enabled IRQ in system VIA.
DD11 6A          j       ROR  A         } A/4
DD12 6A          j       ROR  A         }
DD13 90 54       .T      BCC  &DD69     branch if Bit 1 of Masked byte was 0 (if CA1 interru
DD15 CE 40 02    .@.     DEC  &0240     Decrement CFS timeout counter (is disabled
DD18 A5 EA       ..      LDA  &EA       read CFS counter
DD1A 10 02       ..      BPL  &DD1E     if 0-127 branch
DD1C E6 EA       ..      INC  &EA       & increment CFS counter.
DD1E AD 51 02    .Q.     LDA  &0251     read Flash counter
DD21 F0 1A       ..      BEQ  &DD3D     if 0 branch - this means change the flash colours.
DD23 CE 51 02    .Q.     DEC  &0251     decrement Flash counter.
DD26 D0 15       ..      BNE  &DD3D
DD28 AE 52 02    .R.     LDX  &0252     read mark period count.
DD2B AD 48 02    .H.     LDA  &0248     read Video ULA copy of control register.
DD2E 4A          J       LSR  A         A=A/2
DD2F 90 03       ..      BCC  &DD34     branch if bit 0 of Video ULA copy is 0 (Flash colour bit
DD31 AE 53 02    .S.     LDX  &0253     read space period count.
DD34 2A          *       ROL  A         A=A*2 (restore Accumulator) - same as &26B
DD35 49 01       I.      EOR  #&01      toggle bit 0 - Flash colour select bit
DD37 20 00 EA     ..     JSR  &EA00     write Acc to Video ULA control register (&FE20
DD3A 8E 51 02    .Q.     STX  &0251     Mark period count restored (countdown).
DD3D A0 04       ..      LDY  #&04      } Event 4 call - executes event 4!
DD3F 20 94 E4     ..     JSR  &E494     }
DD42 A9 02       ..      LDA  #&02      A=2 - (bit 1 = 1)
DD44 4C 6E DE    Ln.     JMP  &DE6E     - clear CA1 interrupt flag & exit Interru
DD47 AD 6D FE    .m.     LDA  &FE6D     read User VIA Interrupt flag register.
DD4A 10 A7       ..      BPL  &DCF3     branch if VIA didn't cause interrupt
                                        |- ready to exit!
```

```
DD4C 2D 77 02     -w.    AND  &0277
DD4F 2D 6E FE     -n.    AND  &FE6E
DD52 6A           j      ROR  A
DD53 6A           j      ROR  A
DD54 90 9D        ..     BCC  &DCF3
DD56 AC 85 02     ...    LDY  &0285
DD59 88           .      DEY
DD5A D0 97        ..     BNE  &DCF3
DD5C A9 02        ..     LDA  #&02
DD5E 8D 6D FE     .m.    STA  &FE6D
DD61 8D 6E FE     .n.    STA  &FE6E
DD64 A2 03        ..     LDX  #&03
DD66 4C 3A E1     L:.    JMP  &E13A
DD69 2A           *      ROL  A
DD6A 2A           *      ROL  A
DD6B 2A           *      ROL  A
DD6C 2A           *      ROL  A
DD6D 10 5B        .[     BPL  &DDCA
DD6F A9 20        .      LDA  #&20
DD71 A2 00        ..     LDX  #&00
DD73 8D 4D FE     .M.    STA  &FE4D
DD76 8E 49 FE     .I.    STX  &FE49
DD79 A2 08        ..     LDX  #&08
DD7B 86 FB        ..     STX  &FB
DD7D 20 5B E4      [.    JSR  &E45B
DD80 6E D7 02     n..    ROR  &02D7
DD83 30 44        0D     BMI  &DDC9
DD85 A8           .      TAY
DD86 F0 05        ..     BEQ  &DD8D
DD88 20 6D EE      m.    JSR  &EE6D
DD8B 30 3C        0<     BMI  &DDC9
DD8D 20 60 E4      `.    JSR  &E460
DD90 85 F5        ..     STA  &F5
DD92 20 60 E4      `.    JSR  &E460
DD95 85 F7        ..     STA  &F7
DD97 20 60 E4      `.    JSR  &E460
DD9A 85 F6        ..     STA  &F6
DD9C A4 F5        ..     LDY  &F5
DD9E F0 1B        ..     BEQ  &DDBB
DDA0 10 16        ..     BPL  &DDB8
DDA2 24 F5        $.     BIT  &F5
DDA4 70 05        p.     BVS  &DDAB
DDA6 20 BB EE      ..    JSR  &EEBB
DDA9 50 07        P.     BVC  &DDB2
DDAB 06 F6        ..     ASL  &F6
DDAD 26 F7        &.     ROL  &F7
DDAF 20 3B EE      ;.    JSR  &EE3B
DDB2 AC 61 02     .a.    LDY  &0261
DDB5 4C 7F EE     L..    JMP  &EE7F
DDB8 20 7F EE      ..    JSR  &EE7F
DDBB A4 F6        ..     LDY  &F6
DDBD 20 7F EE      ..    JSR  &EE7F
DDC0 A4 F7        ..     LDY  &F7
DDC2 20 7F EE      ..    JSR  &EE7F
DDC5 46 FB        F.     LSR  &FB
DDC7 D0 B4        ..     BNE  &DD7D
DDC9 60           `      RTS
DDCA 90 7B        .{     BCC  &DE47
DDCC A9 40        .@     LDA  #&40
DDCE 8D 4D FE     .M.    STA  &FE4D
DDD1 AD 83 02     ...    LDA  &0283
DDD4 AA           .      TAX
```

AND with user VIA interrupt mask
AND with interrupts enabled flags
A = A / 4
branch if bit 1 of masking was 0. (exit)
read printer destination flag.
if flag set exit interrupt (more pa
clear CA1 interrupt flag
stop anymore CA1 interrupts.
X=3.
ready

```
DDD5 49 0F       I.     EOR  #&0F
DDD7 48          H      PHA
DDD8 A8          .      TAY
DDD9 BD 91 02    ...    LDA  &0291,X
DDDC 69 00       i.     ADC  #&00
DDDE 99 91 02    ...    STA  &0291,Y
DDE1 CA          .      DEX
DDE2 F0 03       ..     BEQ  &DDE7
DDE4 88          .      DEY
DDE5 D0 F2       ..     BNE  &DDD9
DDE7 68          h      PLA
DDE8 8D 83 02    ...    STA  &0283
DDEB A2 05       ..     LDX  #&05
DDED FE 9B 02    ...    INC  &029B,X
DDF0 D0 08       ..     BNE  &DDFA
DDF2 CA          .      DEX
DDF3 D0 F8       ..     BNE  &DDED
DDF5 A0 05       ..     LDY  #&05
DDF7 20 94 E4     ..    JSR  &E494
DDFA AD B1 02    ...    LDA  &02B1
DDFD D0 08       ..     BNE  &DE07
DDFF AD B2 02    ...    LDA  &02B2
DE02 F0 06       ..     BEQ  &DE0A
DE04 CE B2 02    ...    DEC  &02B2
DE07 CE B1 02    ...    DEC  &02B1
DE0A 2C CE 02    ,..    BIT  &02CE
DE0D 10 0B       ..     BPL  &DE1A
DE0F EE CE 02    ...    INC  &02CE
DE12 58          X      CLI
DE13 20 47 EB     G.    JSR  &EB47
DE16 78          x      SEI
DE17 CE CE 02    ...    DEC  &02CE
DE1A 2C D7 02    ,..    BIT  &02D7
DE1D 30 0C       0.     BMI  &DE2B
DE1F 20 6D EE     m.    JSR  &EE6D
DE22 49 A0       I.     EOR  #&A0
DE24 C9 60       .`     CMP  #&60
DE26 90 03       ..     BCC  &DE2B
DE28 20 79 DD     y.    JSR  &DD79
DE2B 2C B7 D9    ,..    BIT  &D9B7
DE2E 20 A2 DC     ..    JSR  &DCA2
DE31 A5 EC       ..     LDA  &EC
DE33 05 ED       ..     ORA  &ED
DE35 2D 42 02    -B.    AND  &0242
DE38 F0 04       ..     BEQ  &DE3E
DE3A 38          8      SEC
DE3B 20 65 F0     e.    JSR  &F065
DE3E 20 9B E1     ..    JSR  &E19B
DE41 2C C0 FE    ,..    BIT  &FEC0
DE44 70 04       p.     BVS  &DE4A
DE46 60          `      RTS
DE47 2A          *      ROL  A
DE48 10 28       .(     BPL  &DE72
DE4A AE 4C 02    .L.    LDX  &024C
DE4D F0 1D       ..     BEQ  &DE6C
DE4F AD C2 FE    ...    LDA  &FEC2
DE52 9D B5 02    ...    STA  &02B5,X
DE55 AD C1 FE    ...    LDA  &FEC1
DE58 9D B9 02    ...    STA  &02B9,X
DE5B 8E BE 02    ...    STX  &02BE
DE5E A0 03       ..     LDY  #&03
DE60 20 94 E4     ..    JSR  &E494
```

```
DE63 CA          .      DEX
DE64 D0 03       ..     BNE  &DE69
DE66 AE 4D 02    .M.    LDX  &024D
DE69 20 8F DE     ..    JSR  &DE8F
DE6C A9 10       ..     LDA  #&10
DE6E 8D 4D FE    .M.    STA  &FE4D
DE71 60          `      RTS
DE72 2A          *      ROL  A
DE73 2A          *      ROL  A
DE74 2A          *      ROL  A
DE75 2A          *      ROL  A
DE76 10 07       ..     BPL  &DE7F
DE78 20 65 F0     e.    JSR  &F065
DE7B A9 01       ..     LDA  #&01
DE7D D0 EF       ..     BNE  &DE6E
DE7F 4C F3 DC    L..    JMP  &DCF3
DE82 68          h      PLA
DE83 A8          .      TAY
DE84 68          h      PLA
DE85 AA          .      TAX
DE86 68          h      PLA
DE87 85 FC       ..     STA  &FC
DE89 A5 FC       ..     LDA  &FC
DE8B 40          @      RTI
DE8C 8C BE 02    ...    STY  &02BE
DE8F E0 05       ..     CPX  #&05
DE91 90 02       ..     BCC  &DE95
DE93 A2 04       ..     LDX  #&04
DE95 8E 4C 02    .L.    STX  &024C
DE98 AC 4E 02    .N.    LDY  &024E
DE9B 88          .      DEY
DE9C 98          .      TYA
DE9D 29 08       ).     AND  #&08
DE9F 18          .      CLC
DEA0 6D 4C 02    mL.    ADC  &024C
DEA3 E9 00       ..     SBC  #&00
DEA5 8D C0 FE    ...    STA  &FEC0
DEA8 60          `      RTS
DEA9 A9 C3       ..     LDA  #&C3
DEAB 85 FE       ..     STA  &FE
DEAD A9 00       ..     LDA  #&00
DEAF 85 FD       ..     STA  &FD
DEB1 C8          .      INY
DEB2 B1 FD       ..     LDA  (&FD),Y
DEB4 20 E3 FF     ..    JSR  OSASCI
DEB7 AA          .      TAX
DEB8 D0 F7       ..     BNE  &DEB1
DEBA 60          `      RTS
DEBB 8E B1 02    ...    STX  &02B1
DEBE 8C B2 02    ...    STY  &02B2
DEC1 A9 FF       ..     LDA  #&FF
DEC3 D0 02       ..     BNE  &DEC7
DEC5 A9 00       ..     LDA  #&00
DEC7 85 E6       ..     STA  &E6
DEC9 8A          .      TXA
DECA 48          H      PHA
DECB 98          .      TYA
DECC 48          H      PHA
DECD AC 56 02    .V.    LDY  &0256
DED0 F0 14       ..     BEQ  &DEE6
DED2 38          8      SEC
DED3 66 EB       f.     ROR  &EB
```

IRQ2 — restore Acc & exit from inter

*FX 17 Force an ADC conversion

Point to 'BBC Computer'

Message Print routine

OSRDCH

&E6=0, misc 03.

Save X, Y

Any EXEC Files open? branch if not.

Set B7 of &EB — supress message during File read.

```
DED5 20 D7 FF     ..     JSR  OSBGET   read byte from Y File (handle in Y)
DED8 08           .      PHP           save ST (EXEC)
DED9 46 EB        F.     LSR  &EB      Bit 7 of &EB is 0, permit messages -
DEDB 28           (      PLP           retrieve ST
DEDC 90 25        .%     BCC  &DF03    branch if end of file not reached.
DEDE A9 00        ..     LDA  #&00     clear EXEC file active flag.
DEE0 8D 56 02     .V.    STA  &0256
DEE3 20 CE FF     ..     JSR  OSFIND   Close the file.
DEE6 24 FF        $.     BIT  &FF
DEE8 30 16        0.     BMI  &DF00    Branch if Escape has been pressed.
DEEA AE 41 02     .A.    LDX  &0241    read input source number.
DEED 20 77 E5      w.    JSR  &E577    get character from current input stream
DEF0 90 11        ..     BCC  &DF03    C=0 if successful.
DEF2 24 E6        $.     BIT  &E6
DEF4 50 F0        P.     BVC  &DEE6
DEF6 AD B1 02     ...    LDA  &02B1    } Test if 'key' timer has timed out -
DEF9 0D B2 02     ...    ORA  &02B2    } branch if it hasn't
DEFC D0 E8        ..     BNE  &DEE6
DEFE B0 05        ..     BCS  &DF05    always taken (I think)
DF00 38           8      SEC           C=1 indicating 'error'
DF01 A9 1B        ..     LDA  #&1B     A=&1B(27) for escape
DF03 85 E6        ..     STA  &E6      &E6 = byte read from: KBD/Exec file etc.
DF05 68           h      PLA
DF06 A8           .      TAY           Recover Y,X
DF07 68           h      PLA
DF08 AA           .      TAX
DF09 A5 E6        ..     LDA  &E6      A= ASCII of character returned.
DF0B 60           `      RTS           exit.
DF0C 29 43        )C     AND  #&43     '(C)' used to check if ROM is valid (RESET
DF0E 28           (      PLP
DF0F 00           .      BRK
DF10 2E E0 31     ..1    ROL  &31E0
DF13 05 46        .F     ORA  &46
DF15 58           X      CLI
DF16 E3           .      ???
DF17 42           B      ???
DF18 FF           .      ???
DF19 42           B      ???
DF1A 41 53        AS     EOR  (&53,X)
DF1C 49 43        IC     EOR  #&43
DF1E E0 18        ..     CPX  #&18
DF20 00           .      BRK
DF21 43           C      ???
DF22 41 54        AT     EOR  (&54,X)
DF24 E0 31        .1     CPX  #&31
DF26 05 43        .C     ORA  &43
DF28 4F           O      ???
DF29 44           D      ???
DF2A 45 E3        E.     EOR  &E3
DF2C 48           H      PHA
DF2D 88           .      DEY
DF2E 45 58        EX     EOR  &58
DF30 45 43        EC     EOR  &43
DF32 F6 8D        ..     INC  &8D,X
DF34 00           .      BRK
DF35 48           H      PHA
DF36 45 4C        EL     EOR  &4C
DF38 50 F0        P.     BVC  &DF2A
DF3A B9 FF 4B     ..K    LDA  &4BFF,Y
DF3D 45 59        EY     EOR  &59
DF3F E3           .      ???
DF40 27           '      ???
```

* Command table

| | | |
|---|---|---|
| . | &E031 | &05 |
| F X | &E342 | &FF |
| BASIC | &E018 | &00 |
| CAT | &E031 | &05 |
| CODE | &E348 | &88 |
| EXEC | &F68D | &00 |
| HELP | &F0B9 | &FF |
| KEY | &E327 | &FF |

+ Address

locations + data

```
DF41 FF          .      ???
DF42 4C 4F 41    LDA    JMP  &414F
DF45 44          D      ???
DF46 E2          .      ???
DF47 3C          <      ???
DF48 00          .      BRK
DF49 4C 49 4E    LIN    JMP  &4E49
DF4C 45 E6       E.     EOR  &E6
DF4E 59 01 4D    Y.M    EOR  &4D01,Y
DF51 4F          O      ???
DF52 54          T      ???
DF53 4F          O      ???
DF54 52          R      ???
DF55 E3          .      ???
DF56 48          H      PHA
DF57 89          .      ???
DF58 4F          O      ???
DF59 50 54       PT     BVC  &DFAF
DF5B E3          .      ???
DF5C 48          H      PHA
DF5D 8B          .      ???
DF5E 52          R      ???
DF5F 55 4E       UN     EOR  &4E,X
DF61 E0 31       .1     CPX  #&31
DF63 04          .      ???
DF64 52          R      ???
DF65 4F          O      ???
DF66 4D E3 48    M.H    EOR  &48E3
DF69 8D 53 41    .SA    STA  &4153
DF6C 56 45       VE     LSR  &45,X
DF6E E2          .      ???
DF6F 3E 00 53    >.S    ROL  &5300,X
DF72 50 4F       PO     BVC  &DFC3
DF74 4F          O      ???
DF75 4C E2 81    L..    JMP  &81E2
DF78 00          .      BRK
DF79 54          T      ???
DF7A 41 50       AP     EOR  (&50,X)
DF7C 45 E3       E.     EOR  &E3
DF7E 48          H      PHA
DF7F 8C 54 56    .TV    STY  &5654
DF82 E3          .      ???
DF83 48          H      PHA
DF84 90 E0       ..     BCC  &DF66
DF86 31 03       1.     AND  (&03),Y
DF88 00          .      BRK
DF89 86 F2       ..     STX  &F2
DF8B 84 F3       ..     STY  &F3
DF8D A9 08       ..     LDA  #&08
DF8F 20 31 E0     1.    JSR  &E031
DF92 A0 00       ..     LDY  #&00
DF94 B1 F2       ..     LDA  (&F2),Y
DF96 C9 0D       ..     CMP  #&0D
DF98 F0 04       ..     BEQ  &DF9E
DF9A C8          .      INY
DF9B D0 F7       ..     BNE  &DF94
DF9D 60          `      RTS
DF9E A0 FF       ..     LDY  #&FF
DFA0 20 39 E0     9.    JSR  &E039
DFA3 F0 72       .r     BEQ  &E017
DFA5 C9 2A       .*     CMP  #&2A
DFA7 F0 F7       ..     BEQ  &DFA0
```

LOAD &E23C &00
LINE &E659 &01
MOTOR &E348 &89
OPT &E348 &8B
RUN &E031 &04
ROM &E348 &8D
SAVE &E23E &00
SPOOL &E231 &00
TAPE &E348 &8C
TV &E348 &90

locations + Data!

Null command &E031, Param 3. (unrecognised OS command)

End of list

save pointer OSCLI

call 8 to FSCV? not implemented.

check the string has an &D terminator.

exit if no CR (&D)

skip leading spaces - if any.

branch if CR (&D) found - exit

skip extra leading asterisks '*'

```
DFA9 20 3A E0     :.    JSR  &E03A   get next non'space' char, exit if CR(&0D)
DFAC F0 69        .i    BEQ  &E017
DFAE C9 7C        .|    CMP  #&7C    Exit if '*|' command found.
DFB0 F0 65        .e    BEQ  &E017
DFB2 C9 2F        ./    CMP  #&2F    Has a */ (*RUN) command been issued.
DFB4 D0 08        ..    BNE  &DFBE   branch if not
DFB6 C8           .     INY          INC pointer to Start of String
DFB7 20 09 E0     ..    JSR  &E009
DFBA A9 02        ..    LDA  #&02    OSFSC 2 ~ *RUN a file.
DFBC D0 73        .s    BNE  &E031
DFBE 84 E6        ..    STY  &E6     save offset                Star command table search.
DFC0 A2 00        ..    LDX  #&00    table offset = 0
DFC2 F0 13        ..    BEQ  &DFD7
DFC4 5D 10 DF     ]..   EOR  &DF10,X
DFC7 29 DF        ).    AND  #&DF
DFC9 D0 17        ..    BNE  &DFE2   branch if no match.
DFCB C8           .     INY          Next char, C=0 to skip following BCS.
DFCC 18           .     CLC
DFCD B0 25        .%    BCS  &DFF4   C=1 if '.' has terminated the search string.
DFCF E8           .     INX          next char in lookup table.
DFD0 B1 F2        .     LDA  (&F2),Y
DFD2 20 E3 E4     ..    JSR  &E4E3   is it a letter A-Z?
DFD5 90 ED        ..    BCC  &DFC4   branch if it is.
DFD7 BD 10 DF     ...   LDA  &DF10,X get next char of lookup command
DFDA 30 16        0.    BMI  &DFF2   if -ve use byte for execution address
DFDC B1 F2        ..    LDA  (&F2),Y get next char of test string
DFDE C9 2E        ..    CMP  #&2E    is it a '.', if so branch.
DFE0 F0 04        ..    BEQ  &DFE6
DFE2 18           .     CLC          C=0 for NON '.' terminator (See &DFCD)
DFE3 A4 E6        ..    LDY  &E6     Reset Y to Start of test string (DEY cancelled by INY
DFE5 88           .     DEY
DFE6 C8           .     INY
DFE7 E8           .     INX          INX:INX skips execution address
DFE8 E8           .     INX
DFE9 BD 0E DF     ...   LDA  &DF0E,X read next char.
DFEC F0 33        .3    BEQ  &E021   end of command list found
DFEE 10 F8        ..    BPL  &DFE8   branch until -ve number found.
DFF0 30 DB        0.    BMI  &DFCD   always taken
DFF2 E8           .     INX
DFF3 E8           .     INX          } Fudge table pointer appropriately
DFF4 CA           .     DEX
DFF5 CA           .     DEX
DFF6 48           H     PHA          save high of link address.
DFF7 BD 11 DF     ...   LDA  &DF11,X
DFFA 48           H     PHA          Save low  "  "  "
DFFB 20 3A E0     :.    JSR  &E03A   get next non zero char.
DFFE 18           .     CLC          C=0
DFFF 08           .     PHP          Save Status NB Z=1 if <CR> was found!
E000 20 04 E0     ..    JSR  &E004   Z=0 if end not found.
E003 40           @     RTI          call link  X,Y = end of String / next Char.
E004 BD 12 DF     ...   LDA  &DF12,X Read extra parameter byte, exit if -ve
E007 30 0E        0.    BMI  &E017
E009 98           .     TYA          X,Y = end of string.
E00A BC 12 DF     ...   LDY  &DF12,X
E00D 18           .     CLC
E00E 65 F2        e.    ADC  &F2     base pointer low
E010 AA           .     TAX
E011 98           .     TYA          X on exit = low of actual addr
E012 A4 F3        ..    LDY  &F3     Y on exit = high of actual addr
E014 90 01        ..    BCC  &E017
E016 C8           .     INY          (end of string)
E017 60           `     RTS          Exit.
```

```
?D E018 FFFF
E018 AE 4B 02     .K.     LDX  &024B   Socket with BASIC
E01B 30 04        0.      BMI  &E021   NO Basic present *BASIC.
E01D 38           8       SEC
E01E 4C E7 DB     L..     JMP  &DBE7   enter language ROM.
E021 A4 E6        ..      LDY  &E6
E023 A2 04        ..      LDX  #&04    } service call, unrecognised call.
E025 20 68 F1      h.     JSR  &F168   } if ROM accepted it.
E028 F0 ED        ..      BEQ  &E017
E02A A5 E6        ..      LDA  &E6
E02C 20 0D E0      ..     JSR  &E00D
E02F A9 03        ..      LDA  #&03
E031 6C 1E 02     l..     JMP  (&021E)   *CAT, *RUN, *FX139, *FX127
E034 0A           .       ASL  A
E035 29 01        ).      AND  #&01
E037 10 F8        ..      BPL  &E031
E039 C8           .       INY
E03A B1 F2        ..      LDA  (&F2),Y
E03C C9 20        .       CMP  #&20
E03E F0 F9        ..      BEQ  &E039
E040 C9 0D        ..      CMP  #&0D
E042 60           `       RTS
E043 90 F5        ..      BCC  &E03A
E045 20 3A E0      :.     JSR  &E03A
E048 C9 2C        .,      CMP  #&2C
E04A D0 F4        ..      BNE  &E040
E04C C8           .       INY
E04D 60           `       RTS
E04E 20 3A E0      :.     JSR  &E03A
E051 20 7D E0      ).     JSR  &E07D
E054 90 37        .7      BCC  &E08D
E056 85 E6        ..      STA  &E6
E058 20 7C E0      !.     JSR  &E07C
E05B 90 19        ..      BCC  &E076
E05D AA           .       TAX
E05E A5 E6        ..      LDA  &E6
E060 0A           .       ASL  A
E061 B0 2A        .*      BCS  &E08D
E063 0A           .       ASL  A
E064 B0 27        .'      BCS  &E08D
E066 65 E6        e.      ADC  &E6
E068 B0 23        .#      BCS  &E08D
E06A 0A           .       ASL  A
E06B B0 20        .       BCS  &E08D
E06D 85 E6        ..      STA  &E6
E06F 8A           .       TXA
E070 65 E6        e.      ADC  &E6
E072 B0 19        ..      BCS  &E08D
E074 90 E0        ..      BCC  &E056
E076 A6 E6        ..      LDX  &E6
E078 C9 0D        ..      CMP  #&0D
E07A 38           8       SEC
E07B 60           `       RTS
E07C C8           .       INY
E07D B1 F2        ..      LDA  (&F2),Y
E07F C9 3A        .:      CMP  #&3A
E081 B0 0A        ..      BCS  &E08D
E083 C9 30        .0      CMP  #&30
E085 90 06        ..      BCC  &E08D
E087 29 0F        ).      AND  #&0F
E089 60           `       RTS
E08A 20 45 E0      E.     JSR  &E045
```

(FSCV) *OPT equivalent. Check for EOF on an opened File.

text pointer is it a space? branch if it is. is it a CR. exit. (Z=result, Z=1 if CR Z=0 if not).

Skip Spaces & check for CR

Skip spaces (leading) is it a , to next character NB Z=0! unless Y=&FF (wrap)

Skip Spaces and check for comma.

Skip spaces. check if valid decimal digit exit if invalid digit found

EVAL decimal text NO.

Save it. next digit exit from eval routine when non decimal

save A.

(&E6)=(&E6)*10 checks for overflow as well

overflow ... error

recover A

add new digit to result

always taken

X= result of eval.

check if end of text has been found Z=1 if true

C=1 if OK.

Inc offset register text pointer (get a number). colon ? if > or equal branch (CLC & exit) is it < &30 branch if it is (CLC & exit) take bottom 4 bits exit

C=0 if not decimal
C=1 if valid

```
E08D 18          .      CLC      C=0, indicating error in EVAL
E08E 60          `      RTS      exit.
E08F 20 7D E0     }.    JSR  &E07D  get a numeric digit from text (0-9)
E092 B0 0E       ..     BCS  &E0A2  branch if a numeric character was found
E094 29 DF       ).     AND  #&DF   remove bit 5 from character.
E096 C9 47       .G     CMP  #&47   } check if hex digit A-F inclusive.
E098 B0 F0       ..     BCS  &E08A  } if not, branch to &E08A.
E09A C9 41       .A     CMP  #&41
E09C 90 EC       ..     BCC  &E08A
E09E 08          .      PHP
E09F E9 37       .7     SBC  #&37
E0A1 28          (      PLP
E0A2 C8          .      INY
E0A3 60          `      RTS      exit (A = Nibble corresponding to hex digit from t
E0A4 48          H      PHA      Save A          OSWRCH entry point
E0A5 8A          .      TXA      } Save X
E0A6 48          H      PHA      }
E0A7 98          .      TYA      } Save Y
E0A8 48          H      PHA      }
E0A9 BA          .      TSX      X=SP
E0AA BD 03 01    ...    LDA  &0103,X  retrieve Acc from Stack
E0AD 48          H      PHA      push A on Stack, again.
E0AE 2C 60 02    ,`.    BIT  &0260  Econet write character status - bit7=1 if
E0B1 10 08       ..     BPL  &E0BB  branch if not for Econet.   output to
E0B3 A8          .      TAY                                      Econet
E0B4 A9 04       ..     LDA  #&04   Econet only!
E0B6 20 7E E5     ~.    JSR  &E57E
E0B9 B0 52       .R     BCS  &E10D
E0BB 18          .      CLC
E0BC A9 02       ..     LDA  #&02  } check bit 1 - disable VDU driver
E0BE 2C 7C 02    ,|.    BIT  &027C } read write character destination stat
E0C1 D0 05       ..     BNE  &E0C8  branch if VDU driver disabled (*FX3
E0C3 68          h      PLA      } retrieve Acc.
E0C4 48          H      PHA      }
E0C5 20 C0 C4     ..    JSR  &C4C0  — Normal VDU instruction (ie print
E0C8 A9 08       ..     LDA  #&08  } check bit 4 - enable printer bit.
E0CA 2C 7C 02    ,|.    BIT  &027C } character destination status.
E0CD D0 02       ..     BNE  &E0D1
E0CF 90 05       ..     BCC  &E0D6  - See & E0BB
E0D1 68          h      PLA      } retrieve Acc.
E0D2 48          H      PHA      }
E0D3 20 14 E1     ..    JSR  &E114
E0D6 AD 7C 02    .|.    LDA  &027C
E0D9 6A          j      ROR  A   A=A/2 - to test RS423 enable bit (*FX 3).
E0DA 90 1B       ..     BCC  &E0F7  branch if RS423 driver disabled
E0DC A4 EA       ..     LDY  &EA
E0DE 88          .      DEY
E0DF 10 16       ..     BPL  &E0F7
E0E1 68          h      PLA
E0E2 48          H      PHA
E0E3 08          .      PHP
E0E4 78          x      SEI
E0E5 A2 02       ..     LDX  #&02
E0E7 48          H      PHA
E0E8 20 5B E4     [.    JSR  &E45B
E0EB 90 03       ..     BCC  &E0F0
E0ED 20 70 E1     p.    JSR  &E170
E0F0 68          h      PLA
E0F1 A2 02       ..     LDX  #&02
E0F3 20 F8 E1     ..    JSR  &E1F8
E0F6 28          (      PLP
E0F7 A9 10       ..     LDA  #&10  } test spooled output disable flag
                                    1= disabled, 0= enabled.
```

```
E0F9 2C 7C 02     ,|.     BIT  &027C
E0FC D0 0F        ..      BNE  &E10D
E0FE AC 57 02     .W.     LDY  &0257
E101 F0 0A        ..      BEQ  &E10D
E103 68           h       PLA
E104 48           H       PHA
E105 38           8       SEC
E106 66 EB        f.      ROR  &EB
E108 20 D4 FF      ..     JSR  OSBPUT
E10B 46 EB        F.      LSR  &EB
E10D 68           h       PLA
E10E 68           h       PLA
E10F A8           .       TAY
E110 68           h       PLA
E111 AA           .       TAX
E112 68           h       PLA
E113 60           `       RTS
E114 2C 7C 02     ,|.     BIT  &027C
E117 70 20        p       BVS  &E139
E119 CD 86 02     ...     CMP  &0286
E11C F0 1B        ..      BEQ  &E139
E11E 08           .       PHP
E11F 78           x       SEI
E120 AA           .       TAX
E121 A9 04        ..      LDA  #&04
E123 2C 7C 02     ,|.     BIT  &027C
E126 D0 10        ..      BNE  &E138
E128 8A           .       TXA
E129 A2 03        ..      LDX  #&03
E12B 20 F8 E1      ..     JSR  &E1F8
E12E B0 08        ..      BCS  &E138
E130 2C D2 02     ,..     BIT  &02D2
E133 10 03        ..      BPL  &E138
E135 20 3A E1      :.     JSR  &E13A
E138 28           (       PLP
E139 60           `       RTS
E13A AD 85 02     ...     LDA  &0285
E13D F0 6E        .n      BEQ  &E1AD
E13F C9 01        ..      CMP  #&01
E141 D0 21        .!      BNE  &E164
E143 20 60 E4      `.     JSR  &E460
E146 6E D2 02     n..     ROR  &02D2
E149 30 45        0E      BMI  &E190
E14B A0 82        ..      LDY  #&82
E14D 8C 6E FE     .n.     STY  &FE6E
E150 8D 61 FE     .a.     STA  &FE61
E153 AD 6C FE     .l.     LDA  &FE6C
E156 29 F1        ).      AND  #&F1
E158 09 0C        ..      ORA  #&0C
E15A 8D 6C FE     .l.     STA  &FE6C
E15D 09 0E        ..      ORA  #&0E
E15F 8D 6C FE     .l.     STA  &FE6C
E162 D0 2C        .,      BNE  &E190
E164 C9 02        ..      CMP  #&02
E166 D0 29        .)      BNE  &E191
E168 A4 EA        ..      LDY  &EA
E16A 88           .       DEY
E16B 10 40        .@      BPL  &E1AD
E16D 4E D2 02     N..     LSR  &02D2
E170 4E 4F 02     NO.     LSR  &024F
E173 20 41 E7      A.     JSR  &E741
E176 90 18        ..      BCC  &E190
```

branch if spool disabled
- File handle if spool file.
branch if not spooling - exit.
retrieve Acc.
12 & set bit 7 - location &EB
- cassette critical flag.
12 & set bit 7 to 0
- pull copy of Acc from stack
restore Y
restore X
restore A
exit.

```
E178 A2 20        .      LDX  #&20
E17A A0 9F        ..     LDY  #&9F
E17C 08           .      PHP
E17D 78           x      SEI
E17E 98           .      TYA
E17F 86 FA        ..     STX  &FA      general workspace
E181 2D 50 02     -P.    AND  &0250    RS423 control Ploy
E184 45 FA        E.     EOR  &FA      "     "     "
E186 AE 50 02     .P.    LDX  &0250    RS423 control Ploy.
E189 8D 50 02     .P.    STA  &0250    "     "     "
E18C 8D 08 FE     ...    STA  &FE08    6850 control register (RS423).
E18F 28           (      PLP
E190 60           `      RTS

E191 18           .      CLC
E192 A9 01        ..     LDA  #&01
E194 20 A2 E1      ..    JSR  &E1A2
E197 6E D2 02     n..    ROR  &02D2    Printer buffer busy
E19A 60           `      RTS           (Clear bit 7). Ploy.
E19B 2C D2 02     ,..    BIT  &02D2
E19E 30 FA        0.     BMI  &E19A
E1A0 A9 00        ..     LDA  #&00
E1A2 A2 03        ..     LDX  #&03
E1A4 AC 85 02     ...    LDY  &0285
E1A7 20 7E E5      ~.    JSR  &E57E
E1AA 6C 22 02     l".    JMP  (&0222)

E1AD 18           .      CLC
E1AE 48           H      PHA
E1AF 08           .      PHP
E1B0 78           x      SEI
E1B1 B0 08        ..     BCS  &E1BB
E1B3 BD AD E9     ...    LDA  &E9AD,X
E1B6 10 03        ..     BPL  &E1BB
E1B8 20 A2 EC      ..    JSR  &ECA2
E1BB 38           8      SEC
E1BC 7E CF 02     ~..    ROR  &02CF,X
E1BF E0 02        ..     CPX  #&02
E1C1 B0 08        ..     BCS  &E1CB
E1C3 A9 00        ..     LDA  #&00
E1C5 8D 68 02     .h.    STA  &0268
E1C8 8D 6A 02     .j.    STA  &026A
E1CB 20 3B E7      ;.    JSR  &E73B
E1CE 28           (      PLP
E1CF 68           h      PLA
E1D0 60           `      RTS

E1D1 50 07        P.     BVC  &E1DA
E1D3 BD D8 02     ...    LDA  &02D8,X
E1D6 9D E1 02     ...    STA  &02E1,X
E1D9 60           `      RTS
E1DA 08           .      PHP
E1DB 78           x      SEI
E1DC 08           .      PHP
E1DD 38           8      SEC
E1DE BD E1 02     ...    LDA  &02E1,X
E1E1 FD D8 02     ...    SBC  &02D8,X
E1E4 B0 04        ..     BCS  &E1EA
E1E6 38           8      SEC
E1E7 FD 47 E4     .G.    SBC  &E447,X
E1EA 28           (      PLP
E1EB 90 06        ..     BCC  &E1F3
E1ED 18           .      CLC
E1EE 7D 47 E4     }G.    ADC  &E447,X
E1F1 49 FF        I.     EOR  #&FF
```

*FX156 Read/Update 6850 ACIA control register & 0.5 copy

general workspace
RS423 control Ploy
" " "
RS423 control Ploy.
6850 control register (RS423).

Printer buffer busy *FX123 Inform 0.5 & Printer (Clear bit 7). Ploy. driver going dormant.

used by VDU2 (&C546

save A, ST and disable IRQs.

2.

set top bit of (&2CF+X) (buffer busy)
Index ≥ 2 then branch. (see OSBYTE21)
A = 0
Length of string = 0 (!?)
VDU queue = 0 (!?)
buffer count/purge routine call & JMP(222E
Restore ST, A, then exit.

OSCNPV count/Purge buffer vect
buffer start indices
buffer end indices, purge buffer (X=buffer numbe

count buffer

buffer end indices
buffer start indices.

```
E1F3 A0 00      ..      LDY  #&00
E1F5 AA         .       TAX
E1F6 28         (       PLP
E1F7 60         `       RTS
E1F8 78         x       SEI
E1F9 20 B0 E4     ..    JSR  &E4B0
E1FC 90 0F      ..      BCC  &E20D
E1FE 20 EA E9     ..    JSR  &E9EA
E201 08         .       PHP
E202 48         H       PHA
E203 20 EB EE     ..    JSR  &EEEB
E206 68         h       PLA
E207 28         (       PLP
E208 30 03      0.      BMI  &E20D
E20A 58         X       CLI
E20B B0 EB      ..      BCS  &E1F8
E20D 60         `       RTS
E20E 48         H       PHA
E20F A9 00      ..      LDA  #&00
E211 9D EE 02   ...     STA  &02EE,X
E214 9D EF 02   ...     STA  &02EF,X
E217 9D F0 02   ...     STA  &02F0,X
E21A 9D F1 02   ...     STA  &02F1,X
E21D 68         h       PLA
E21E 60         `       RTS
E21F 84 E6      ..      STY  &E6
E221 2A         *       ROL  A
E222 2A         *       ROL  A
E223 2A         *       ROL  A
E224 2A         *       ROL  A
E225 A0 04      ..      LDY  #&04
E227 2A         *       ROL  A
E228 3E EE 02   >..     ROL  &02EE,X
E22B 3E EF 02   >..     ROL  &02EF,X
E22E 3E F0 02   >..     ROL  &02F0,X
E231 3E F1 02   >..     ROL  &02F1,X
E234 B0 31      .1      BCS  &E267
E236 88         .       DEY
E237 D0 EE      ..      BNE  &E227
E239 A4 E6      ..      LDY  &E6
E23B 60         `       RTS
E23C A9 FF      ..      LDA  #&FF
E23E 86 F2      ..      STX  &F2
E240 84 F3      ..      STY  &F3
E242 8E EE 02   ...     STX  &02EE
E245 8C EF 02   ...     STY  &02EF
E248 48         H       PHA
E249 A2 02      ..      LDX  #&02
E24B 20 0E E2     ..    JSR  &E20E
E24E A0 FF      ..      LDY  #&FF
E250 8C F4 02   ...     STY  &02F4
E253 C8         .       INY
E254 20 1D EA     ..    JSR  &EA1D
E257 20 2F EA     /.    JSR  &EA2F
E25A 90 FB      ..      BCC  &E257
E25C 68         h       PLA
E25D 48         H       PHA
E25E F0 62      .b      BEQ  &E2C2
E260 20 AD E2     ..    JSR  &E2AD
E263 B0 3B      .;      BCS  &E2A0
E265 F0 3E      .>      BEQ  &E2A5
E267 00         .       BRK
```

PUT Acc into buffer, X=buffer number. (C=1 & fail) if OK.

- try until buffer succeeds to request.

- save A.

Store Ø in OSFILE Control block area

Xreg = offset.

restore A.

general workspace. (text offset pointer usually the lower nibble into upper nibble (carry is usually set) at start.

Shift osfile control block bytes left 4 times

- bad address.

restore text pointer offset

exit.

*LOAD entry point (A=&FF).

*SAVE

Text Pointer

copy of above.

Save Acc.

&2FØ-&2F3 incl = Ø.

Y=Ø

check Filename is OK.

Branch if save (AC=Ø)

get load address from text.

branch if load address is valid.

never taken ??

'Bad address'

```
E268 FC          .      ???
E269 42          B      ???
E26A 61 64       ad     ADC  (&64,X)
E26C 20 61 64     ad    JSR  &6461
E26F 64          d      ???
E270 72          r      ???
E271 65 73       es     ADC  &73
E273 73          s      ???
E274 00          .      BRK
E275 A2 10       ..     LDX  #&10
E277 20 68 F1     h.    JSR  &F168
E27A F0 23       .#     BEQ  &E29F
E27C 20 8B F6     ..    JSR  &F68B
E27F A9 00       ..     LDA  #&00
E281 08          .      PHP
E282 84 E6       ..     STY  &E6
E284 AC 57 02    .W.    LDY  &0257
E287 8D 57 02    .W.    STA  &0257
E28A F0 03       ..     BEQ  &E28F
E28C 20 CE FF     ..    JSR  OSFIND
E28F A4 E6       ..     LDY  &E6
E291 28          (      PLP
E292 F0 0B       ..     BEQ  &E29F
E294 A9 80       ..     LDA  #&80
E296 20 CE FF     ..    JSR  OSFIND
E299 A8          .      TAY
E29A F0 74       .t     BEQ  &E310
E29C 8D 57 02    .W.    STA  &0257
E29F 60          `      RTS
E2A0 D0 6E       .n     BNE  &E310
E2A2 EE F4 02    ...    INC  &02F4
E2A5 A2 EE       ..     LDX  #&EE
E2A7 A0 02       ..     LDY  #&02
E2A9 68          h      PLA
E2AA 4C DD FF    L..    JMP  OSFILE
E2AD 20 3A E0     :.    JSR  &E03A
E2B0 20 8F E0     ..    JSR  &E08F
E2B3 90 0C       ..     BCC  &E2C1
E2B5 20 0E E2     ..    JSR  &E20E
E2B8 20 1F E2     ..    JSR  &E21F
E2BB 20 8F E0     .     JSR  &E08F
E2BE B0 F8       ..     BCS  &E2B8
E2C0 38          8      SEC
E2C1 60          `      RTS
E2C2 A2 0A       ..     LDX  #&0A
E2C4 20 AD E2     ..    JSR  &E2AD
E2C7 90 47       .G     BCC  &E310
E2C9 B8          .      CLV
E2CA B1 F2       ..     LDA  (&F2),Y
E2CC C9 2B       .+     CMP  #&2B
E2CE D0 04       ..     BNE  &E2D4
E2D0 2C B7 D9    ,..    BIT  &D9B7
E2D3 C8          .      INY
E2D4 A2 0E       ..     LDX  #&0E
E2D6 20 AD E2     ..    JSR  &E2AD
E2D9 90 35       .5     BCC  &E310
E2DB 08          .      PHP
E2DC 50 0F       P.     BVC  &E2ED
E2DE A2 FC       ..     LDX  #&FC
E2E0 18          .      CLC
E2E1 BD FC 01    ...    LDA  &01FC,X
E2E4 7D 00 02    }..    ADC  &0200,X
```

'Bad address'

&FC

*FX 119 Close any SPOOL or EXEC files.

OSFSC A=6

*SPOOL

File handle of current SPOOL.

File handle of current SPOOL.

Bad command (Invalid text).

point to control block at &2EE

— Acc Ø = Save &FF = load.

(control block has been completed From *LOAD *SAVE

check for CR. Load routine

get & test for a hex digit — From text pointer (&F2,&F3)

branch if not a hex digit

Store Ø's in OSFile control block.

build up load/save address } Ø Ø.

get next digit

C=1 = successful (valid line)

on exit C=Ø = error

Save routine

```
E2E7 9D 00 02     ...    STA  &0200,X
E2EA E8           .      INX
E2EB D0 F4        ..     BNE  &E2E1
E2ED A2 03        ..     LDX  #&03
E2EF BD F8 02     ...    LDA  &02F8,X
E2F2 9D F4 02     ...    STA  &02F4,X
E2F5 9D F0 02     ...    STA  &02F0,X
E2F8 CA           .      DEX
E2F9 10 F4        ..     BPL  &E2EF
E2FB 28           (      PLP
E2FC F0 A7        ..     BEQ  &E2A5
E2FE A2 06        ..     LDX  #&06
E300 20 AD E2      ..    JSR  &E2AD
E303 90 0B        ..     BCC  &E310
E305 F0 9E        ..     BEQ  &E2A5
E307 A2 02        ..     LDX  #&02
E309 20 AD E2      ..    JSR  &E2AD
E30C 90 02        ..     BCC  &E310
E30E F0 95        ..     BEQ  &E2A5
E310 00           .      BRK
E311 FE 42 61     .Ba    INC  &6142,X
E314 64           d      ???
E315 20 63 6F      co    JSR  &6F63
E318 6D 6D 61     mma    ADC  &616D
E31B 6E 64 00     nd.    ROR  &0064
E31E FB           .      ???
E31F 42           B      ???
E320 61 64        ad     ADC  (&64,X)
E322 20 6B 65      ke    JSR  &656B
E325 79 00 20     y.     ADC  &2000,Y
E328 4E E0 90     N..    LSR  &90E0
E32B F1 E0        ..     SBC  (&E0),Y
E32D 10 B0        ..     BPL  &E2DF
E32F ED 20 45     . E    SBC  &4520
E332 E0 08        ..     CPX  #&08
E334 AE 10 0B     ...    LDX  &0B10
E337 98           .      TYA
E338 48           H      PHA
E339 20 D1 E3      ..    JSR  &E3D1
E33C 68           h      PLA
E33D A8           .      TAY
E33E 28           (      PLP
E33F D0 36        .6     BNE  &E377
E341 60           `      RTS
E342 20 4E E0      N.    JSR  &E04E
E345 90 C9        ..     BCC  &E310
E347 8A           .      TXA
E348 48           H      PHA
E349 A9 00        ..     LDA  #&00
E34B 85 E5        ..     STA  &E5
E34D 85 E4        ..     STA  &E4
E34F 20 43 E0      C.    JSR  &E043
E352 F0 18        ..     BEQ  &E36C
E354 20 4E E0      N.    JSR  &E04E
E357 90 B7        ..     BCC  &E310
E359 86 E5        ..     STX  &E5
E35B 20 45 E0      E.    JSR  &E045
E35E F0 0C        ..     BEQ  &E36C
E360 20 4E E0      N.    JSR  &E04E
E363 90 AB        ..     BCC  &E310
E365 86 E4        ..     STX  &E4
E367 20 3A E0      :.    JSR  &E03A
```

*CODE / *LINE  OSRSCV A=3.

'Bad command' &FE

'Bad Key' &FB

*KEY

JSR &E04E – read number
BCC &E31D – bad number
CPX #&10
BCS &E31D – Key no > 15 (error)
JSR &E045 – check for CR
PHP

Save Y

Recover Y

branch if text follows *KEY'n'...

exit.

*FX

See OSCLI

Save A.

*CODE, *MOTOR, *OPT, *ROM, *TAPE, *TV

=0 (X&Y params for *TV, osbyte = 0.)

skip leading spaces and

if ... execute relevant call to osbyte

Evaluate decimal text number.

Bad command

X = 1st param of osbyte.

– skip and check for comma.

branch if ... found.

EVALuate Y parameter of OSCLI string.

– Bad command.

Save result.

Skip spaces (trailing) & check for <CR>

```
?D E36A FFFF
E36A D0 A4        ..      BNE   &E310     - 'Bad command'
E36C A4 E4        ..      LDY   &E4       Read X and Y parameters for the call
E36E A6 E5        ..      LDX   &E5
E370 68           h       PLA             recover osbyte call number, (from CLI)
E371 20 F4 FF      ..     JSR   OSBYTE    execute it.
E374 70 9A        p.      BVS   &E310     If error 'Bad command' will follow
E376 60           `       RTS             exit OSCLI
E377 38           8       SEC             C=1
E378 20 1E EA      ..     JSR   &EA1E
E37B 20 2F EA      /.     JSR   &EA2F
E37E B0 08        ..      BCS   &E388
E380 E8           .       INX
E381 F0 9A        ..      BEQ   &E31D     - 'Bad Key' (No room left to add string)
E383 9D 00 0B     ...     STA   &0B00,X
E386 90 F3        ..      BCC   &E37B
E388 D0 93        ..      BNE   &E31D     - 'Bad Key'
E38A 08           .       PHP             Save ST, NO IRQs
E38B 78           x       SEI
E38C 20 D1 E3      ..     JSR   &E3D1
E38F A2 10        ..      LDX   #&10      X=&10 (loops down to 0 inclusive)
E391 E4 E6        ..      CPX   &E6
E393 F0 0E        ..      BEQ   &E3A3
E395 BD 00 0B     ...     LDA   &0B00,X
E398 D9 00 0B     ...     CMP   &0B00,Y
E39B D0 06        ..      BNE   &E3A3
E39D AD 10 0B     ...     LDA   &0B10
E3A0 9D 00 0B     ...     STA   &0B00,X
E3A3 CA           .       DEX
E3A4 10 EB        ..      BPL   &E391
E3A6 28           (       PLP             Recover ST & Exit
E3A7 60           `       RTS
E3A8 08           .       PHP
E3A9 78           x       SEI
E3AA AD 10 0B     ...     LDA   &0B10
E3AD 38           8       SEC
E3AE F9 00 0B     ...     SBC   &0B00,Y
E3B1 85 FB        ..      STA   &FB
E3B3 8A           .       TXA
E3B4 48           H       PHA
E3B5 A2 10        ..      LDX   #&10
E3B7 BD 00 0B     ...     LDA   &0B00,X
E3BA 38           8       SEC
E3BB F9 00 0B     ...     SBC   &0B00,Y
E3BE 90 08        ..      BCC   &E3C8
E3C0 F0 06        ..      BEQ   &E3C8
E3C2 C5 FB        ..      CMP   &FB
E3C4 B0 02        ..      BCS   &E3C8
E3C6 85 FB        ..      STA   &FB
E3C8 CA           .       DEX
E3C9 10 EC        ..      BPL   &E3B7
E3CB 68           h       PLA
E3CC AA           .       TAX
E3CD A5 FB        ..      LDA   &FB
E3CF 28           (       PLP
E3D0 60           `       RTS
E3D1 08           .       PHP
E3D2 78           x       SEI
E3D3 8A           .       TXA
E3D4 48           H       PHA
E3D5 A4 E6        ..      LDY   &E6
```

```
E3D7 20 A8 E3     ..      JSR  &E3A8
E3DA B9 00 0B     ...     LDA  &0B00,Y
E3DD A8           .       TAY
E3DE 18           .       CLC
E3DF 65 FB        e.      ADC  &FB
E3E1 AA           .       TAX
E3E2 85 FA        ..      STA  &FA
E3E4 AD 68 02     .h.     LDA  &0268
E3E7 F0 0D        ..      BEQ  &E3F6
E3E9 00           .       BRK
E3EA FA           .       ???
E3EB 4B           K       ???
E3EC 65 79        ey      ADC  &79
E3EE 20 69 6E      in     JSR  &6E69
E3F1 20 75 73      us     JSR  &7375
E3F4 65 00        e.      ADC  &00
E3F6 CE 84 02     ...     DEC  &0284
E3F9 68           h       PLA
E3FA 38           8       SEC
E3FB E5 FA        ..      SBC  &FA
E3FD 85 FA        ..      STA  &FA
E3FF F0 0C        ..      BEQ  &E40D
E401 BD 01 0B     ...     LDA  &0B01,X
E404 99 01 0B     ...     STA  &0B01,Y
E407 C8           .       INY
E408 E8           .       INX
E409 C6 FA        ..      DEC  &FA
E40B D0 F4        ..      BNE  &E401
E40D 98           .       TYA
E40E 48           H       PHA
E40F A4 E6        ..      LDY  &E6
E411 A2 10        ..      LDX  #&10
E413 BD 00 0B     ...     LDA  &0B00,X
E416 D9 00 0B     ...     CMP  &0B00,Y
E419 90 07        ..      BCC  &E422
E41B F0 05        ..      BEQ  &E422
E41D E5 FB        ..      SBC  &FB
E41F 9D 00 0B     ...     STA  &0B00,X
E422 CA           .       DEX
E423 10 EE        ..      BPL  &E413
E425 AD 10 0B     ...     LDA  &0B10
E428 99 00 0B     ...     STA  &0B00,Y
E42B 68           h       PLA
E42C 8D 10 0B     ...     STA  &0B10
E42F AA           .       TAX
E430 EE 84 02     ...     INC  &0284
E433 28           (       PLP
E434 60           `       RTS
E435 03           .       ???
E436 0A           .       ASL  A
E437 08           .       PHP
E438 07           .       ???
E439 07           .       ???
E43A 07           .       ???
E43B 07           .       ???
E43C 07           .       ???
E43D 09 00        ..      ORA  #&00
E43F 00           .       BRK
E440 C0 C0        ..      CPY  #&C0
E442 50 60        P`      BVC  &E4A4
E444 70 80        p.      BVS  &E3C6
E446 00           .       BRK
```

'key in use' &FA

3 00
A 00
8 00
7 C0
7 50
760
770
780
900

high of ~~input~~ base buffer Addresses.

~~Data~~ ?

low of ~~input~~ base buffer addresses.
(9 buffers)

```
E447 E0 00      ..      CPX  #&00      base offset
E449 40         @       RTI            for buffers &E1E7.
E44A C0 F0      ..      CPY  #&F0      (See previous tables)
E44C F0 F0      ..      BEQ  &E43E
E44E F0 C0      ..      BEQ  &E410
E450 BD 3E E4   .>.     LDA  &E43E,X   } read pointer into &FA for
E453 85 FA      ..      STA  &FA       } base address of buffer given in X
E455 BD 35 E4   .5.     LDA  &E435,X   }  (see *FX 21 for values).
E458 85 FB      ..      STA  &FB
E45A 60         `       RTS

E45B 2C B7 D9   ,..     BIT  &D9B7     *FX 152 Examine Buffer Status.
E45E 70 01      p.      BVS  &E461
E460 B8         .       CLV            *FX 145 Get character from buffer
E461 6C 2C 02   l,.     JMP  (&022C)   (&22C) = &E464 (OS 1.20).

E464 08         .       PHP            OSREM
E465 78         x       SEI  No IRQs
E466 BD D8 02   ...     LDA  &02D8,X   buffer start index.
E469 DD E1 02   ...     CMP  &02E1,X   buffer end index.
E46C F0 72      .r      BEQ  &E4E0     if buffer empty C=1 and exit. (by taking the branch)
E46E A8         .       TAY            Y= start index.
E46F 20 50 E4    P.     JSR  &E450     read base address of selected buffer
E472 B1 FA      ..      LDA  (&FA),Y   read byte from buffer
E474 70 1B      p.      BVS  &E491     depends upon entry point if &E464 then V=0!
E476 48         H       PHA            Save value on Stack
E477 C8         .       INY            increment pointer
E478 98         .       TYA            A= pointer
E479 D0 03      ..      BNE  &E47E     if pointer not reached max value branch
E47B BD 47 E4   .G.     LDA  &E447,X   get start value for selected buffer.
E47E 9D D8 02   ...     STA  &02D8,X   save pointer.
E481 E0 02      ..      CPX  #&02      If input stream = 0 or 1 then branch (KBD or RS423 input)
E483 90 0A      ..      BCC  &E48F
E485 DD E1 02   ...     CMP  &02E1,X   If start & End don't match then branch (branch taken if buffer not empty)
E488 D0 05      ..      BNE  &E48F
E48A A0 00      ..      LDY  #&00      Issue event 0 (in Y)
E48C 20 94 E4    ..     JSR  &E494
E48F 68         h       PLA            recover byte read from buffer.
E490 A8         .       TAY            into Y
E491 28         (       PLP            recover ST.
E492 18         .       CLC            C=0, for successfull exit
E493 60         `       RTS            the end

E494 08         .       PHP            Save SR        OSEVEN (Jump to event if it is enabled).
E495 78         x       SEI            D=1
E496 48         H       PHA            Save A
E497 85 FA      ..      STA  &FA       - general OS workspace.
E499 B9 BF 02   ...     LDA  &02BF,Y   read event flag - (0= event disabled, <>7 0= enabled)
E49C F0 41      .A      BEQ  &E4DF     - disabled for event in Y neg.
E49E 98         .       TYA            A= event number
E49F A4 FA      ..      LDY  &FA       Y= old A's value.
E4A1 20 A5 F0    ..     JSR  &F0A5     - jump through event vector.
E4A4 68         h       PLA            restore A
E4A5 28         (       PLP            restore Status reg.
E4A6 18         .       CLC            C=0; indicating event was issued and enabled.
E4A7 60         `       RTS            return.

E4A8 98         .       TYA            A= character tried to be inserted into buffer.
E4A9 A0 02      ..      LDY  #&02      } Issue event 2 - character entering input buffer
E4AB 20 94 E4    ..     JSR  &E494     }
E4AE A8         .       TAY            Y= character being used.
E4AF 98         .       TYA            to A.          *FX 138 Insert value into buffer.
E4B0 6C 2A 02   l*.     JMP  (&022A)   &E4B3 OS 1.20.

E4B3 08         .       PHP            Save ST.       OSINS
E4B4 78         x       SEI            No IRQs.
E4B5 48         H       PHA            Save A (byte being placed in selected buffer)
```

```
E4B6 BC E1 02     ...    LDY  &02E1,X
E4B9 C8           .      INY
E4BA D0 03        ..     BNE  &E4BF
E4BC BC 47 E4     .G.    LDY  &E447,X
E4BF 98           .      TYA
E4C0 DD D8 02     ...    CMP  &02D8,X
E4C3 F0 0F        ..     BEQ  &E4D4
E4C5 BC E1 02     ...    LDY  &02E1,X
E4C8 9D E1 02     ...    STA  &02E1,X
E4CB 20 50 E4      P.    JSR  &E450
E4CE 68           h      PLA
E4CF 91 FA        ..     STA  (&FA),Y
E4D1 28           (      PLP
E4D2 18           .      CLC
E4D3 60           `      RTS
E4D4 68           h      PLA
E4D5 E0 02        ..     CPX  #&02
E4D7 B0 07        ..     BCS  &E4E0
E4D9 A0 01        ..     LDY  #&01
E4DB 20 94 E4      ..    JSR  &E494
E4DE 48           H      PHA
E4DF 68           h      PLA
E4E0 28           (      PLP
E4E1 38           8      SEC
E4E2 60           `      RTS
E4E3 48           H      PHA
E4E4 29 DF        ).     AND  #&DF
E4E6 C9 41        .A     CMP  #&41
E4E8 90 04        ..     BCC  &E4EE
E4EA C9 5B        .[     CMP  #&5B
E4EC 90 01        ..     BCC  &E4EF
E4EE 38           8      SEC
E4EF 68           h      PLA
E4F0 60           `      RTS
E4F1 A2 00        ..     LDX  #&00
E4F3 8A           .      TXA
E4F4 2D 45 02     -E.    AND  &0245
E4F7 D0 B6        ..     BNE  &E4AF
E4F9 98           .      TYA
E4FA 4D 6C 02     M1.    EOR  &026C
E4FD 0D 75 02     .u.    ORA  &0275
E500 D0 A6        ..     BNE  &E4A8
E502 AD 58 02     .X.    LDA  &0258
E505 6A           j      ROR  A
E506 98           .      TYA
E507 B0 0A        ..     BCS  &E513
E509 A0 06        ..     LDY  #&06
E50B 20 94 E4      ..    JSR  &E494
E50E 90 03        ..     BCC  &E513
E510 20 74 E6      t.    JSR  &E674
E513 18           .      CLC
E514 60           `      RTS
E515 6A           j      ROR  A
E516 68           h      PLA
E517 B0 79        .y     BCS  &E592
E519 98           .      TYA
E51A 48           H      PHA
E51B 4A           J      LSR  A
E51C 4A           J      LSR  A
E51D 4A           J      LSR  A
E51E 4A           J      LSR  A
E51F 49 04        I.     EOR  #&04
```

read buffer end index.
+1
branch if not wrapped round end of buffer.
read start offset for buffer.
A= offset for character being removed.
Branch if its the same as the start index.
read end index
save new index
read buffer base address into &FA,&FB.
recover byte to be inserted into buffer.
put in buffer.
recover ST.
C=0 for success.
exit.
recover value being inserted into buffer.
If buffer ≥2 then exit
Issue event 1 - Input buffer full.
unsuccessful exit routine used several times. C=1 for bad execution
Recover ST.
C=1
exit
to Upper case
<'A', branch
C=0 if its a letter
else C=1 if invalid
check if A contains a letter 'A-Z'
on exit C=1 if invalid
C=0 if a letter
A= buffer (0 or 1) *FX 153
RS423 mode
branch if escapes/spaces/event are ignored (calls OSBYTE)
A= character being inserted into buffer.
Escape character.
escape key status (0 if normal action)
Issue event 2 - character entering buffer, + put in buffer
Escape disabled flag.
A= character to be placed in buffer.
Issue event 6 - Escape condition detected.
branch if event was enabled and issued.
set bit 7 of &FF & JMP to &4003 if tube present
C=0, exit.
Insert character into input buffer, checking Escape.

```
E521 A8          .      TAY
E522 B9 65 02    .e.    LDA  &0265,Y
E525 C9 01       ..     CMP  #&01
E527 F0 6B       .k     BEQ  &E594
E529 68          h      PLA
E52A 90 0D       ..     BCC  &E539
E52C 29 0F       ).     AND  #&0F
E52E 18          .      CLC
E52F 79 65 02    ye.    ADC  &0265,Y
E532 18          .      CLC
E533 60          `      RTS
E534 20 6F E8     o.    JSR  &E86F
E537 68          h      PLA
E538 AA          .      TAX
E539 20 60 E4     `.    JSR  &E460
E53C B0 55       .U     BCS  &E593
E53E 48          H      PHA
E53F E0 01       ..     CPX  #&01
E541 D0 06       ..     BNE  &E549
E543 20 73 E1     s.    JSR  &E173
E546 A2 01       ..     LDX  #&01
E548 38          8      SEC
E549 68          h      PLA
E54A 90 05       ..     BCC  &E551
E54C AC 45 02    .E.    LDY  &0245
E54F D0 41       .A     BNE  &E592
E551 A8          .      TAY
E552 10 3E       .>     BPL  &E592
E554 29 0F       ).     AND  #&0F
E556 C9 0B       ..     CMP  #&0B
E558 90 BF       ..     BCC  &E519
E55A 69 7B       i{     ADC  #&7B
E55C 48          H      PHA
E55D AD 7D 02    .}.    LDA  &027D
E560 D0 B3       ..     BNE  &E515
E562 AD 7C 02    .|.    LDA  &027C
E565 6A          j      ROR  A
E566 6A          j      ROR  A
E567 68          h      PLA
E568 B0 CF       ..     BCS  &E539
E56A C9 87       ..     CMP  #&87
E56C F0 38       .8     BEQ  &E5A6
E56E A8          .      TAY
E56F 8A          .      TXA
E570 48          H      PHA
E571 98          .      TYA
E572 20 CE D8     ..    JSR  &D8CE
E575 68          h      PLA
E576 AA          .      TAX
E577 2C 5F 02    ,_.    BIT  &025F
E57A 10 05       ..     BPL  &E581
E57C A9 06       ..     LDA  #&06
E57E 6C 24 02    l$.    JMP  (&0224)
E581 AD 68 02    .h.    LDA  &0268
E584 F0 B3       ..     BEQ  &E539
E586 AC C9 02    ...    LDY  &02C9
E589 B9 01 0B    ...    LDA  &0B01,Y
E58C EE C9 02    ...    INC  &02C9
E58F CE 68 02    .h.    DEC  &0268
E592 18          .      CLC
E593 60          `      RTS
E594 68          h      PLA
```

get character from buffer (x defines input stream)
exit with C=1 if buffer was empty.
save A?
Is Input stream RS423, branch if not.
read repeat rate, branch if <>0 for successful exit.
Y = char.
successfull exit
branch if econet isn't the input source.
get character from Econet.
read length of soft key string.
branch if empty
read soft key expansion pointer.
read character from key definitions.
inc offset pointer
DEC length of soft key string.
C=0 for successful, then exit

```
E595 29 0F        ).      AND  #&0F
E597 A8           .       TAY
E598 20 A8 E3      ..     JSR  &E3A8
E59B 8D 68 02     .h.     STA  &0268
E59E B9 00 0B     ...     LDA  &0B00,Y
E5A1 8D C9 02     ...     STA  &02C9
E5A4 D0 D1        ..      BNE  &E577
E5A6 8A           .       TXA
E5A7 48           H       PHA
E5A8 20 05 D9      ..     JSR  &D905
E5AB A8           .       TAY
E5AC F0 86        ..      BEQ  &E534
E5AE 68           h       PLA
E5AF AA           .       TAX
E5B0 98           .       TYA
E5B1 18           .       CLC
E5B2 60           `       RTS
E5B3 21 E8        !.      AND  (&E8,X)
E5B5 88           .       DEY
E5B6 E9 D3        ..      SBC  #&D3
E5B8 E6 97        ..      INC  &97
E5BA E9 97        ..      SBC  #&97
E5BC E9 76        .v      SBC  #&76
E5BE E9 88        ..      SBC  #&88
E5C0 E9 8B        ..      SBC  #&8B
E5C2 E6 89        ..      INC  &89
E5C4 E6 B0        ..      INC  &B0
E5C6 E6 B2        ..      INC  &B2
E5C8 E6 95        ..      INC  &95
E5CA E9 8C        ..      SBC  #&8C
E5CC E9 F9        ..      SBC  #&F9
E5CE E6 FA        ..      INC  &FA
E5D0 E6 A8        ..      INC  &A8
E5D2 F0 06        ..      BEQ  &E5DA
E5D4 E7           .       ???
E5D5 8C DE C8     ...     STY  &C8DE
E5D8 E9 B6        ..      SBC  #&B6
E5DA E9 07        ..      SBC  #&07
E5DC CD B4 F0     ...     CMP  &F0B4
E5DF 6C E8 D9     l..     JMP  (&D9E8)
E5E2 E9 75        .u      SBC  #&75
E5E4 E2           .       ???
E5E5 45 F0        E.      EOR  &F0
E5E7 CF           .       ???
E5E8 F0 CD        ..      BEQ  &E5B7
E5EA F0 97        ..      BEQ  &E583
E5EC E1 73        .s      SBC  (&73,X)
E5EE E6 74        .t      INC  &74
E5F0 E6 5C        .\      INC  &5C
E5F2 E6 35        .5      INC  &35
E5F4 E0 4F        .O      CPX  #&4F
E5F6 E7           .       ???
E5F7 13           .       ???
E5F8 E7           .       ???
E5F9 29 E7        ).      AND  #&E7
E5FB 85 F0        ..      STA  &F0
E5FD 23           #       ???
E5FE D9 26 D9     .&.     CMP  &D926,Y
E601 47           G       ???
E602 D6 C2        ..      DEC  &C2,X
E604 D7           .       ???
E605 57           W       ???
```

OSBYTE locations low, high. + OSWORD at end.

```
E606 E6 7F        ..      INC   &7F
E608 E6 AF        ..      INC   &AF
E60A E4 34        .4      CPX   &34
E60C E0 35        .5      CPX   #&35
E60E F1 35        .5      SBC   (&35),Y
E610 F1 E7        ..      SBC   (&E7),Y
E612 DB           .       ???
E613 68           h       PLA
E614 F1 E3        ..      SBC   (&E3),Y
E616 EA           .       NOP
E617 60           `       RTS
E618 E4 AA        ..      CPX   &AA
E61A FF           .       ???
E61B F4           .       ???
E61C EA           .       NOP
E61D AE FF F9     ...     LDX   &F9FF
E620 EA           .       NOP
E621 B2           .       ???
E622 FF           .       ???
E623 FE EA 5B     ..[     INC   &5BEA,X
E626 E4 F3        ..      CPX   &F3
E628 E4 FF        ..      CPX   &FF
E62A E9 10        ..      SBC   #&10
E62C EA           .       NOP
E62D 7C           |       ???
E62E E1 A7        ..      SBC   (&A7,X)
E630 FF           .       ???
E631 6D EE 7F     m..     ADC   &7FEE
E634 EE C0 E9     ...     INC   &E9C0
E637 9C           .       ???
E638 E9 59        .Y      SBC   #&59
E63A E6 02        ..      INC   &02
E63C E9 D5        ..      SBC   #&D5
E63E E8           .       INX
E63F E8           .       INX
E640 E8           .       INX
E641 D1 E8        ..      CMP   (&E8),Y
E643 E4 E8        ..      CPX   &E8
E645 03           .       ???
E646 E8           .       INX
E647 0B           .       ???
E648 E8           .       INX
E649 2D E8 AE     -..     AND   &AEE8
E64C E8           .       INX
E64D 35 C7        5.      AND   &C7,X
E64F F3           .       ???
E650 CB           .       ???
E651 48           H       PHA
E652 C7           .       ???
E653 E0 C8        ..      CPX   #&C8
E655 CE D5 A9     ...     DEC   &A9D5
E658 00           .       BRK
E659 6C 00 02     l..     JMP   (&0200)
E65C A2 00        ..      LDX   #&00
E65E 24 FF        $.      BIT   &FF
E660 10 11        ..      BPL   &E673
E662 AD 76 02     .v.     LDA   &0276
E665 D0 0A        ..      BNE   &E671
E667 58           X       CLI
E668 8D 69 02     .i.     STA   &0269
E66B 20 8D F6      ..     JSR   &F68D
E66E 20 AA F0      ..     JSR   &F0AA
```

Action Address!

*FX 136 *CODE equivalent.

*LINE

*FX 126 Acknowledge detection of an Escape condition.

```
E671 A2 FF        ..      LDX  #&FF
E673 18           .       CLC
E674 66 FF        f.      ROR  &FF
E676 2C 7A 02     ,z.     BIT  &027A
E679 30 01        0.      BMI  &E67C
E67B 60           `       RTS
E67C 4C 03 04     L..     JMP  &0403
E67F AD 82 02     ...     LDA  &0282
E682 A8           .       TAY
E683 2A           *       ROL  A
E684 E0 01        ..      CPX  #&01
E686 6A           j       ROR  A
E687 50 1E        P.      BVC  &E6A7
E689 A9 38        .8      LDA  #&38
E68B 49 3F        I?      EOR  #&3F
E68D 85 FA        ..      STA  &FA
E68F AC 82 02     ...     LDY  &0282
E692 E0 09        ..      CPX  #&09
E694 B0 17        ..      BCS  &E6AD
E696 3D AD E9     =..     AND  &E9AD,X
E699 85 FB        ..      STA  &FB
E69B 98           .       TYA
E69C 05 FA        ..      ORA  &FA
E69E 45 FA        E.      EOR  &FA
E6A0 05 FB        ..      ORA  &FB
E6A2 09 40        .@      ORA  #&40
E6A4 4D 5D 02     M].     EOR  &025D
E6A7 8D 82 02     ...     STA  &0282
E6AA 8D 10 FE     ...     STA  &FE10
E6AD 98           .       TYA
E6AE AA           .       TAX
E6AF 60           `       RTS
E6B0 C8           .       INY
E6B1 18           .       CLC
E6B2 B9 52 02     .R.     LDA  &0252,Y
E6B5 48           H       PHA
E6B6 8A           .       TXA
E6B7 99 52 02     .R.     STA  &0252,Y
E6BA 68           h       PLA
E6BB A8           .       TAY
E6BC AD 51 02     .Q.     LDA  &0251
E6BF D0 10        ..      BNE  &E6D1
E6C1 8E 51 02     .Q.     STX  &0251
E6C4 AD 48 02     .H.     LDA  &0248
E6C7 08           .       PHP
E6C8 6A           j       ROR  A
E6C9 28           (       PLP
E6CA 2A           *       ROL  A
E6CB 8D 48 02     .H.     STA  &0248
E6CE 8D 20 FE     . .     STA  &FE20
E6D1 50 DA        P.      BVC  &E6AD
E6D3 8A           .       TXA
E6D4 29 01        ).      AND  #&01
E6D6 48           H       PHA
E6D7 AD 50 02     .P.     LDA  &0250
E6DA 2A           *       ROL  A
E6DB E0 01        ..      CPX  #&01
E6DD 6A           j       ROR  A
E6DE CD 50 02     .P.     CMP  &0250
E6E1 08           .       PHP
E6E2 8D 50 02     .P.     STA  &0250
E6E5 8D 08 FE     ...     STA  &FE08
```

*FX124 Clear Escape condition.
*FX125 set " " .

*FX137 *MOTOR equivalent

*FX8 set RS423 baud rate For TX.
*FX7 set RS423 baud rate For RX.

serial processor (ULA) routine.
'write'.

*FX9 set duration For Flashing Cols

*FX10 " " " " "

*FX 2 select input Stream.

```
E6E8 20 73 E1      s.      JSR  &E173
E6EB 28            (       PLP
E6EC F0 03         ..      BEQ  &E6F1
E6EE 2C 09 FE      ,..     BIT  &FE09
E6F1 AE 41 02      .A.     LDX  &0241
E6F4 68            h       PLA
E6F5 8D 41 02      .A.     STA  &0241
E6F8 60            `       RTS
```

*FX13 Disable events.
*FX14 Enable events.

```
E6F9 98            .       TYA
E6FA E0 0A         ..      CPX  #&0A
E6FC B0 B0         ..      BCS  &E6AE
E6FE BC BF 02      ...     LDY  &02BF,X
E701 9D BF 02      ...     STA  &02BF,X
E704 50 A7         P.      BVC  &E6AD
```

*FX16 Select ADC channels to be sampled.

```
E706 F0 03         ..      BEQ  &E70B
E708 20 8C DE       ..     JSR  &DE8C
E70B AD 4D 02      .M.     LDA  &024D
E70E 8E 4D 02      .M.     STX  &024D
E711 AA            .       TAX
E712 60            `       RTS
```

*FX129 Read key with time limit.

```
E713 98            .       TYA
E714 30 0B         0.      BMI  &E721
E716 58            X       CLI
E717 20 BB DE       ..     JSR  &DEBB
E71A B0 03         ..      BCS  &E71F
E71C AA            .       TAX
E71D A9 00         ..      LDA  #&00
E71F A8            .       TAY
E720 60            `       RTS
```

Branch if -ve inkey operation.

-ve inkey routine.

```
E721 8A            .       TXA
E722 49 7F         I.      EOR  #&7F
E724 AA            .       TAX
E725 20 68 F0       h.     JSR  &F068
E728 2A            *       ROL  A
```

*FX130 Read machine high order Address.

```
E729 A2 FF         ..      LDX  #&FF
E72B A0 FF         ..      LDY  #&FF
E72D B0 02         ..      BCS  &E731
E72F E8            .       INX
E730 C8            .       INY
E731 60            `       RTS
E732 8A            .       TXA
E733 49 FF         I.      EOR  #&FF
E735 AA            .       TAX
E736 E0 02         ..      CPX  #&02
E738 B8            .       CLV
E739 50 03         P.      BVC  &E73E
E73B 2C B7 D9      ,..     BIT  &D9B7
E73E 6C 2E 02      l..     JMP  (&022E)
E741 38            8       SEC
E742 A2 01         ..      LDX  #&01
E744 20 38 E7       8.     JSR  &E738
E747 C0 01         ..      CPY  #&01
E749 B0 03         ..      BCS  &E74E
E74B EC 5B 02      .[.     CPX  &025B
E74E 60            `       RTS
```

N=1, V=1
buffer count/purge vector.

*FX128 Read ADC channel or get buffer status.

```
E74F 30 E1         0.      BMI  &E732
E751 F0 0C         ..      BEQ  &E75F
E753 E0 05         ..      CPX  #&05
E755 B0 D2         ..      BCS  &E729
E757 BC B9 02      ...     LDY  &02B9,X
E75A BD B5 02      ...     LDA  &02B5,X
E75D AA            .       TAX
```

```
E75E 60          `      RTS
E75F AD 40 FE    .@.    LDA  &FE40
E762 6A          j      ROR  A
E763 6A          j      ROR  A
E764 6A          j      ROR  A
E765 6A          j      ROR  A
E766 49 FF       I.     EOR  #&FF
E768 29 03       ).     AND  #&03
E76A AC BE 02    ...    LDY  &02BE
E76D 8E BE 02    ...    STX  &02BE
E770 AA          .      TAX
E771 60          `      RTS
E772 48          H      PHA  Save osbyte Num OSBYTE entry point
E773 08          .      PHP  save ST
E774 78          x      SEI  NO IRQs.
E775 85 EF       ..     STA  &EF   Save A,X&Y
E777 86 F0       ..     STX  &F0
E779 84 F1       ..     STY  &F1
E77B A2 07       ..     LDX  #&07  X=7
E77D C9 75       .u     CMP  #&75  OSBYTE <= &74 branch if so (0-&74)
E77F 90 41       .A     BCC  &E7C2
E781 C9 A1       ..     CMP  #&A1
E783 90 09       ..     BCC  &E78E  OSBYTE >= &75 & < &A1 branch if so (&75-&
E785 C9 A6       ..     CMP  #&A6
E787 90 3F       .?     BCC  &E7C8  OSBYTE >= &A1 & < &A6 branch if (&A1-&
E789 18          .      CLC        C=0
E78A A9 A1       ..     LDA  #&A1
E78C 69 00       i.     ADC  #&00  &A2 -> Acc (when C=0)
E78E 38          8      SEC
E78F E9 5F       ._     SBC  #&5F
E791 0A          .      ASL  A     A=A*2, C=1 (A=&24 when C=0)
E792 38          8      SEC
E793 84 F1       ..     STY  &F1   Save Y param of OSBYTE/OSWORD
E795 A8          .      TAY        Y= table offset
E796 2C 5E 02    ,^.    BIT  &025E Econet intercept status (-ve for econet control)
E799 10 07       ..     BPL  &E7A2
E79B 8A          .      TXA
E79C B8          .      CLV
E79D 20 7E E5     ~.    JSR  &E57E
E7A0 70 1A       p.     BVS  &E7BC
E7A2 B9 B4 E5    ...    LDA  &E5B4,Y
E7A5 85 FB       ..     STA  &FB
E7A7 B9 B3 E5    ...    LDA  &E5B3,Y
E7AA 85 FA       ..     STA  &FA
E7AC A5 EF       ..     LDA  &EF   A,Y = original OSBYTE parameter values
E7AE A4 F1       ..     LDY  &F1
E7B0 B0 04       ..     BCS  &E7B6 take unless Econet (I think)
E7B2 A0 00       ..     LDY  #&00
E7B4 B1 F0       ..?    LDA  (&F0),Y  Read byte pointed to by X and Y values.
E7B6 38          8      SEC
E7B7 A6 F0       ..     LDX  &F0   X= value from OSBYTE call.
E7B9 20 58 F0     X.    JSR  &F058 JMP (&FA) - execute the OSBYTE.
E7BC 6A          j      ROR  A     A/2
E7BD 28          (      PLP        recover ST.
E7BE 2A          *      ROL  A     A*2 - use 'C' from initial ST!
E7BF 68          h      PLA        recover AC.
E7C0 B8          .      CLV        V=0
E7C1 60          `      RTS        exit OSBYTE routine. (Valid OSBYTE Exit)
E7C2 A0 00       ..     LDY  #&00  Y=0
E7C4 C9 16       ..     CMP  #&16  is OSBYTE < 16
E7C6 90 C9       ..     BCC  &E791 branch if 0 to &15 (valid)
E7C8 08          .      PHP        Save ST for invalid call.
```

OSBYTE is &A6–&F
R/W OS variables

OSBYTEs 0 to &

Bad OSBYTE/OSWORD call

```
E7C9 08          .      PHP          and again!
E7CA 68          h      PLA          Retrieve them.
E7CB 68          h      PLA
E7CC 20 68 F1     h.    JSR  &F168   issue service call
E7CF D0 05       ..     BNE  &E7D6   branch if not accepted
E7D1 A6 F0       ..     LDX  &F0     X=original osbyte, JUMP to accepted OS
E7D3 4C BC E7    L..    JMP  &E7BC
E7D6 28          (      PLP          Recover ST, A.
E7D7 68          h      PLA
E7D8 2C B7 D9    ,..    BIT  &D9B7   V=1, N=1
E7DB 60          `      RTS          Exit OSBYTE
E7DC A5 EB       ..     LDA  &EB     cassette critical flag.
E7DE 30 32       02     BMI  &E812   -> A=0 & exit.
E7E0 A9 08       ..     LDA  #&08
E7E2 25 E2       %.     AND  &E2     cassette FS status byte.
E7E4 D0 04       ..     BNE  &E7EA
E7E6 A9 88       ..     LDA  #&88
E7E8 25 BB       %.     AND  &BB
E7EA 60          `      RTS
E7EB 48          H      PHA  Save A, ST    OSWORD entry point
E7EC 08          .      PHP
E7ED 78          x      SEI  IRQs off.
E7EE 85 EF       ..     STA  &EF
E7F0 86 F0       ..     STX  &F0     save original A,X&Y
E7F2 84 F1       ..     STY  &F1
E7F4 A2 08       ..     LDX  #&08
E7F6 C9 E0       ..     CMP  #&E0    Branch if OSWORD >= &E0, X=8 for service call if
E7F8 B0 90       ..     BCS  &E78A   OSWORD rejected.
E7FA C9 0E       ..     CMP  #&0E
E7FC B0 CA       ..     BCS  &E7C8   Branch if OSWORD >= &E (Invalid call has been ma
E7FE 69 44       iD     ADC  #&44    +&44
E800 0A          .      ASL  A       *2
E801 90 90       ..     BCC  &E793   always taken - this executes the OSWORD link rout
E803 20 15 E8     ..    JSR  &E815   OSWORD 5  Read I/O processor
E806 A1 F9       ..     LDA  (&F9,X) read byte and save at          memory.
E808 91 F0       ..     STA  (&F0),Y and of control block.
E80A 60          `      RTS
E80B 20 15 E8     ..    JSR  &E815   OSWORD 6  Write I/O processor
E80E B1 F0       ..     LDA  (&F0),Y                               memory.
E810 81 F9       ..     STA  (&F9,X) write parameter byte.
E812 A9 00       ..     LDA  #&00
E814 60          `      RTS
E815 85 FA       ..     STA  &FA     save 1st byte of the osword block
E817 C8          .      INY          INC offset
E818 B1 F0       ..     LDA  (&F0),Y get 2nd byte in control block.
E81A 85 FB       ..     STA  &FB     - save address (high)
E81C A0 04       ..     LDY  #&04    - offset into osword block.
E81E A2 01       ..     LDX  #&01    - " for destination
E820 60          `      RTS          exit.
E821 D0 FB       ..     BNE  &E81E   X<>0, *FX0 Identify OS version
E823 00          .      BRK          branch is time to return X=1
E824 F7          .      ???          BRK 'OS 1.20' &F7
E825 4F          O      ???
E826 53          S      ???
E827 20 31 2E     1.    JSR  &2E31
E82A 32          2      ???
E82B 30 00       0.     BMI  &E82D
E82D C8          .      INY          OSWORD 7 'sound' command
E82E B1 F0       ..     LDA  (&F0),Y read channel number high byte
E830 C9 FF       ..     CMP  #&FF    if FFxx then it's for the speech processor, so branch.
E832 F0 59       .Y     BEQ  &E88D
E834 C9 20       .      CMP  #&20    channel >=&20?
```

```
E836 A2 08      ..      LDX  #&08      If it is issue service call 8 (X=8)
E838 B0 90      ..      BCS  &E7CA
E83A 88         .       DEY            dec offset                (channel low byte)
E83B 20 C9 E8     ..    JSR  &E8C9     } get byte from control block, ANDed with 3
E83E 09 04      ..      ORA  #&04      } then set bit 2 and place in X. (X between 4 & 7
E840 AA         .       TAX
E841 90 05      ..      BCC  &E848     branch if the previous byte fetched was < &10
E843 20 AE E1     ..    JSR  &E1AE     NB C is now set! Set buffer busy flag & process(?)
E846 A0 01      ..      LDY  #&01      Read channel number high byte (AND #3)
E848 20 C9 E8     ..    JSR  &E8C9     and save in &FA.
E84B 85 FA      ..      STA  &FA
E84D 08         .       PHP            save ST, NB (set if channel high >=&10.
E84E A0 06      ..      LDY  #&06
E850 B1 F0      ..      LDA  (&F0),Y   Read Duration low & push.
E852 48         H       PHA
E853 A0 04      ..      LDY  #&04
E855 B1 F0      ..      LDA  (&F0),Y   Read Pitch low & push.
E857 48         H       PHA
E858 A0 02      ..      LDY  #&02      Read Amplitude low
E85A B1 F0      ..      LDA  (&F0),Y
E85C 2A         *       ROL  A         *2
E85D 38         8       SEC
E85E E9 02      ..      SBC  #&02      -2
E860 0A         .       ASL  A
E861 0A         .       ASL  A         *4
E862 05 FA      ..      ORA  &FA       OR with (channel number high AND #3)
E864 20 F8 E1     ..    JSR  &E1F8     enter into buffer (??)
E867 90 1E      ..      BCC  &E887     appears to be always taken.
E869 68         h       PLA            } Remove parameter bytes, read VDU status & exit
E86A 68         h       PLA            }
E86B 28         (       PLP            }
E86C A6 D0      ..      LDX  &D0       *FX117 Read VDU status.
E86E 60         `       RTS
E86F 08         .       PHP            VDU 7
E870 78         x       SEI
E871 AD 63 02   .c.     LDA  &0263     read BELL (CTRL-G) channel.
E874 29 07      ).      AND  #&07      get bits 0-2.
E876 09 04      ..      ORA  #&04      set bit 2.
E878 AA         .       TAX            A=>X
E879 AD 64 02   .d.     LDA  &0264     Amplitude/Envelope for BELL.
E87C 20 B0 E4     ..    JSR  &E4B0     insert value into buffer.
E87F AD 66 02   .f.     LDA  &0266     BELL duration.
E882 48         H       PHA            save
E883 AD 65 02   .e.     LDA  &0265     BELL frequency.
E886 48         H       PHA            save
E887 38         8       SEC            C=1. ROR &800 + channel (set bit
E888 7E 00 08   ~..     ROR  &0800,X   -
E88B 30 17      0.      BMI  &E8A4     - ~~if -ve~~, this branch is always taken
E88D 08         .       PHP            save ST
E88E C8         .       INY
E88F B1 F0      ..      LDA  (&F0),Y   read block from X&Y reg.
E891 48         H       PHA            } save & increment offset.
E892 C8         .       INY
E893 B1 F0      ..      LDA  (&F0),Y   read block from X&Y reg.
E895 48         H       PHA            save byte
E896 A0 00      ..      LDY  #&00
E898 B1 F0      ..      LDA  (&F0),Y   read block from X&Y reg.
E89A A2 08      ..      LDX  #&08      } put byte into speech buffer.
E89C 20 F8 E1     ..    JSR  &E1F8     -
E89F B0 C8      ..      BCS  &E869     - exit, after removing parameters.
E8A1 6E D7 02   n..     ROR  &02D7     - buffer busy flag for speech process
E8A4 68         h       PLA            recover pitch (low) from stack.
```

```
E8A5 20 B0 E4    ..     JSR  &E4B0   - insert value into buffer
E8A8 68          h      PLA          recover duration(low)
E8A9 20 B0 E4    ..     JSR  &E4B0   - insert value into buffer
E8AC 28          (      PLP          restore ST.
E8AD 60          `      RTS          - exit From OSWORD(7) Call.
E8AE E9 01       ..     SBC  #&01    -1 (envelope OSWORD 8 Define an envelope.
E8B0 0A          .      ASL  A         number in A on entry) Cis also set to (1)
E8B1 0A          .      ASL  A       } *16
E8B2 0A          .      ASL  A
E8B3 0A          .      ASL  A
E8B4 09 0F       ..     ORA  #&0F    set bits 0-3 to %1111 (15).
E8B6 AA          .      TAX          A=>X.  envelope num = (EN-1) *16 + 15
E8B7 A9 00       ..     LDA  #&00      if top 4 bits= envelope bottom 4 bits all set
E8B9 A0 10       ..     LDY  #&10    NB Params &E to &10 always 0!
E8BB C0 0E       ..     CPY  #&0E    } if parameter-&E or above Zero it
E8BD B0 02       ..     BCS  &E8C1   }
E8BF B1 F0       ..     LDA  (&F0),Y read next envelope parameter
E8C1 9D C0 08    ...    STA  &08C0,X store in envelope data area.
E8C4 CA          .      DEX
E8C5 88          .      DEY          } copy 16 bytes into ENVELOPE area.
E8C6 D0 F3       ..     BNE  &E8BB   }
E8C8 60          `      RTS          exit From OSWORD 8
E8C9 B1 F0       ..     LDA  (&F0),Y get next byte in OSWORD control block.
E8CB C9 10       ..     CMP  #&10    CMP with 16.
E8CD 29 03       ).     AND  #&03    take bits 0 & 1 only.
E8CF C8          .      INY          inc osword offset pointer
E8D0 60          `      RTS          Exit (Z always 0)
E8D1 A2 0F       ..     LDX  #&0F    OSWORD 3 Read interval timer.
E8D3 D0 03       ..     BNE  &E8D8
E8D5 AE 83 02    ...    LDX  &0283   read current interval clock base address (5 or 10)   OSWORD 1 Read system clock.
E8D8 A0 04       ..     LDY  #&04    } Copy system clock time to OSWORD block (5 bytes)
E8DA BD 8D 02    ...    LDA  &028D,X }
E8DD 91 F0       ..     STA  (&F0),Y }
E8DF E8          .      INX          }
E8E0 88          .      DEY          }
E8E1 10 F7       ..     BPL  &E8DA   }
E8E3 60          `      RTS          exit osword 1
E8E4 A9 0F       ..     LDA  #&0F    OSWORD 4 write interval timer
E8E6 D0 06       ..     BNE  &E8EE
E8E8 AD 83 02    ...    LDA  &0283   Read which clock OSWORD 2 Write system clock
E8EB 49 0F       I.     EOR  #&0F    select other clock
E8ED 18          .      CLC          C=0
E8EE 48          H      PHA          save base offset for other clock
E8EF AA          .      TAX          =>X
E8F0 A0 04       ..     LDY  #&04    } Copy 5 bytes to clock.
E8F2 B1 F0       ..     LDA  (&F0),Y }
E8F4 9D 8D 02    ...    STA  &028D,X }
E8F7 E8          .      INX          }
E8F8 88          .      DEY          }
E8F9 10 F7       ..     BPL  &E8F2   }
E8FB 68          h      PLA          recover clock base offset.
E8FC B0 E5       ..     BCS  &E8E3   Only exit if middle of routine was called
E8FE 8D 83 02    ...    STA  &0283   change to other system clock.
E901 60          `      RTS          exit OSWORD 2
E902 A0 04       ..     LDY  #&04    OSWORD 0 Read line From input.
E904 B1 F0       ..     LDA  (&F0),Y } Copy bytes 4 => 2 to &2B3 => &2B5 inclusive
E906 99 B1 02    ...    STA  &02B1,Y }
E909 88          .      DEY          }
E90A C0 02       ..     CPY  #&02    }
E90C B0 F6       ..     BCS  &E904   }
E90E B1 F0       ..     LDA  (&F0),Y
E910 85 E9       ..     STA  &E9     &E9 = buffer high byte
```

&2B3 = Max line length
&2B4 = Min Ascii value
&2B5 = Max Ascii value

```
E912 88          .      DEY                 Y now 0
E913 8C 69 02    .i.    STY  &0269          &269=0 always
E916 B1 F0       ..     LDA  (&F0),Y        &E8=low of buffer address
E918 85 E8       ..     STA  &E8
E91A 58          X      CLI
E91B 90 07       ..     BCC  &E924          always taken
E91D A9 07       ..     LDA  #&07           VDU 7 which is given when line too long
E91F 88          .      DEY                 } Accept/reject pointer alter
E920 C8          .      INY                 }
E921 20 EE FF      ..   JSR  OSWRCH         Print the character
E924 20 E0 FF      ..   JSR  OSRDCH         read another character
E927 B0 49       .I     BCS  &E972          branch if Escape was pressed. (C=1)
E929 AA          .      TAX                 X=character received } or error condition.
E92A AD 7C 02    .!.    LDA  &027C          read character destination (*FX 3)
E92D 6A          j      ROR  A              } test bit 1, set if VDU driver disabled.
E92E 6A          j      ROR  A              }
E92F 8A          .      TXA                 A=character from X.
E930 B0 05       ..     BCS  &E937
E932 AE 6A 02    .j.    LDX  &026A          Number of items in VDU queue.
E935 D0 EA       ..     BNE  &E921          Branch if not empty (print it).
E937 C9 7F       ..     CMP  #&7F           delete? branch if not.
E939 D0 07       ..     BNE  &E942
E93B C0 00       ..     CPY  #&00           If no characters have been entered ignore
E93D F0 E5       ..     BEQ  &E924                the delete.
E93F 88          .      DEY                 Decrement number of characters read and
E940 B0 DF       ..     BCS  &E921                always take the branch.
E942 C9 15       ..     CMP  #&15           Is it CTRL-U (clear line), branch if not.
E944 D0 0D       ..     BNE  &E953
E946 98          .      TYA
E947 F0 DB       ..     BEQ  &E924          If no characters have been entered ignore
E949 A9 7F       ..     LDA  #&7F                 the VDU 21 and read another character.
E94B 20 EE FF      ..   JSR  OSWRCH         } Perform n' deletes to erase text then
E94E 88          .      DEY                 }
E94F D0 FA       ..     BNE  &E94B          } Start from the beginning!
E951 F0 D1       ..     BEQ  &E924          always taken.
E953 91 E8       ..     STA  (&E8),Y        save character in buffer area given by control bl
E955 C9 0D       ..     CMP  #&0D           (CR?), branch to exit routine if it is.
E957 F0 13       ..     BEQ  &E96C
E959 CC B3 02    ...    CPY  &02B3          Is it >= to the maximum line length, brow
E95C B0 BF       ..     BCS  &E91D                if limit reached (VDU 7)
E95E CD B4 02    ...    CMP  &02B4          less than min char? branch if it is (reject if
E961 90 BC       ..     BCC  &E91F                taken
E963 CD B5 02    ...    CMP  &02B5          equal to Max? branch if true (accept last c
E966 F0 B8       ..     BEQ  &E920          less than Max Ascii, so accept it.
E968 90 B6       ..     BCC  &E920          reject it.
E96A B0 B3       ..     BCS  &E91F          Move cursor to next line
E96C 20 E7 FF      ..   JSR  OSNEWL         Read JMP (8224) - Econet vector (misc)
E96F 20 7E E5      ~.   JSR  &E57E          Escape flag
E972 A5 FF       ..     LDA  &FF            *2 (C=bit 7)
E974 2A          *      ROL  A              Exit OSWORD 0 (C=1 if Escape pressed else 0
E975 60          `      RTS
E976 58          X      CLI                 ON/OFF JBQS   *FX5  select printer destination.
E977 78          x      SEI
E978 24 FF       $.     BIT  &FF            Branch if Escape pressed.
E97A 30 30       00     BMI  &E9AC
E97C 2C D2 02    ,..    BIT  &02D2          Branch if printer buffer buffer not empty
E97F 10 F5       ..     BPL  &E976
E981 20 A4 E1      ..   JSR  &E1A4          econet vector call, call user printer routine
E984 A0 00       ..     LDY  #&00           &F1=0
E986 84 F1       ..     STY  &F1
E988 09 F0       ..     ORA  #&F0           always  *FX1   Read/write the user Flag
E98A D0 0E       ..     BNE  &E99A          taken.  *FX6   Printer ignore character
```

```
?D E98C FFFF
E98C D0 07        ..      BNE   &E995
E98E A2 32        .2      LDX   #&32
E990 8E 54 02     .T.     STX   &0254
E993 A2 08        ..      LDX   #&08
E995 69 CF        i.      ADC   #&CF
E997 18           .       CLC
E998 69 E9        i.      ADC   #&E9
E99A 86 F0        ..      STX   &F0
E99C A8           .       TAY
E99D B9 90 01     ...     LDA   &0190,Y
E9A0 AA           .       TAX
E9A1 25 F1        %.      AND   &F1
E9A3 45 F0        E.      EOR   &F0
E9A5 99 90 01     ...     STA   &0190,Y
E9A8 B9 91 01     ...     LDA   &0191,Y
E9AB A8           .       TAY
E9AC 60           `       RTS
E9AD 64           d       ???
E9AE 7F           .       ???
E9AF 5B           [       ???
E9B0 6D C9 F6     m..     ADC   &F6C9
E9B3 D2           .       ???
E9B4 E4 40        .@      CPX   &40
E9B6 AD 40 02     .@.     LDA   &0240
E9B9 58           X       CLI
E9BA 78           x       SEI
E9BB CD 40 02     .@.     CMP   &0240
E9BE F0 F9        ..      BEQ   &E9B9
E9C0 BC 01 03     ...     LDY   &0301,X
E9C3 BD 00 03     ...     LDA   &0300,X
E9C6 AA           .       TAX
E9C7 60           `       RTS
E9C8 A9 10        ..      LDA   #&10
E9CA 8D 84 02     ...     STA   &0284
E9CD A2 00        ..      LDX   #&00
E9CF 9D 00 0B     ...     STA   &0B00,X
E9D2 E8           .       INX
E9D3 D0 FA        ..      BNE   &E9CF
E9D5 8E 84 02     ...     STX   &0284
E9D8 60           `       RTS
E9D9 08           .       PHP
E9DA 78           x       SEI
E9DB A9 40        .@      LDA   #&40
E9DD 20 EA E9      ..     JSR   &E9EA
E9E0 30 05        0.      BMI   &E9E7
E9E2 18           .       CLC
E9E3 B8           .       CLV
E9E4 20 68 F0      h.     JSR   &F068
E9E7 28           (       PLP
E9E8 2A           *       ROL   A
E9E9 60           `       RTS
E9EA 90 09        ..      BCC   &E9F5
E9EC A0 07        ..      LDY   #&07
E9EE 8C 40 FE     .@.     STY   &FE40
E9F1 88           .       DEY
E9F2 8C 40 FE     .@.     STY   &FE40
E9F5 24 FF        $.      BIT   &FF
E9F7 60           `       RTS
E9F8 08           .       PHP
E9F9 78           x       SEI
```

take if new delay <> 0

New delay time

*FX12 set auto-repeat rate

*FX11 Set auto-repeat delay.

*FX3 Select output stream.

*FX4 Enable/Disable cursor keys

All OSBYTE CALLS ≥ 166 (&A6) Pass through here.

X = previous value

write value

Y = following OSBYTE value !

exit (several) OSBYTE routines

4 5 6 7

See &E1B3 (sound)

CFS timeout counter

*FX19 wait for Vertical Sync.

Frame counter

if same, branch.

*FX160 Read VDU variable table.

*FX18 Reset soft keys.

soft key consistency flag = &10

&B00 – &BFF = &10

soft key consistency flag = 0

*FX 118 Reflect Keyboard Status in LED's.

'test escape flag'

branch if 'escape' pressed.

JMP (keyboard control vector)

exit

=7

=6

} test 'Escape' flag and exit.

```
E9FA 8D 40 FE     .@.     STA  &FE40
E9FD 28           (       PLP
E9FE 60           `       RTS
E9FF 8A           .       TXA                    *FX 154   write to Video ULA
EA00 08           .       PHP save SR                      control register.
EA01 78           x       SEI disable interrupts -
EA02 8D 48 02     .H.     STA  &0248 - copies of Video ULA control & OS copy.
EA05 8D 20 FE     . .     STA  &FE20 - Video ULA control register.
EA08 AD 53 02     .S.     LDA  &0253 - space period counter.
EA0B 8D 51 02     .Q.     STA  &0251 - Flash counter
EA0E 28           (       PLP restore SR.
EA0F 60           `       RTS exit.
EA10 8A           .       TXA                    *FX 155   write to Video palette reg.
EA11 49 07        I.      EOR  #&07 invert(B0-2)           & OS copy.
EA13 08           .       PHP  save ST
EA14 78           x       SEI  NO IRQs.
EA15 8D 49 02     .I.     STA  &0249 OS copy of Video ULA registers
EA18 8D 21 FE     .!.     STA  &FE21 write to palette.
EA1B 28           (       PLP  Recover ST
EA1C 60           `       RTS  exit.
EA1D 18           .       CLC
EA1E 66 E4        f.      ROR  &E4                  GSINIT
EA20 20 3A E0      :.     JSR  &E03A
EA23 C8           .       INY
EA24 C9 22        ."      CMP  #&22
EA26 F0 02        ..      BEQ  &EA2A
EA28 88           .       DEY
EA29 18           .       CLC
EA2A 66 E4        f.      ROR  &E4
EA2C C9 0D        ..      CMP  #&0D
EA2E 60           `       RTS
EA2F A9 00        ..      LDA  #&00                 GSREAD
EA31 85 E5        ..      STA  &E5
EA33 B1 F2        ..      LDA  (&F2),Y
EA35 C9 0D        ..      CMP  #&0D
EA37 D0 06        ..      BNE  &EA3F
EA39 24 E4        $.      BIT  &E4
EA3B 30 52        0R      BMI  &EA8F
EA3D 10 1B        ..      BPL  &EA5A
EA3F C9 20        .       CMP  #&20
EA41 90 4C        .L      BCC  &EA8F
EA43 D0 06        ..      BNE  &EA4B
EA45 24 E4        $.      BIT  &E4
EA47 30 40        0@      BMI  &EA89
EA49 50 0F        P.      BVC  &EA5A
EA4B C9 22        ."      CMP  #&22
EA4D D0 10        ..      BNE  &EA5F
EA4F 24 E4        $.      BIT  &E4
EA51 10 36        .6      BPL  &EA89
EA53 C8           .       INY
EA54 B1 F2        ..      LDA  (&F2),Y
EA56 C9 22        ."      CMP  #&22
EA58 F0 2F        ./      BEQ  &EA89
EA5A 20 3A E0      :.     JSR  &E03A
EA5D 38           8       SEC
EA5E 60           `       RTS
EA5F C9 7C        .|      CMP  #&7C
EA61 D0 26        .&      BNE  &EA89
EA63 C8           .       INY
EA64 B1 F2        ..      LDA  (&F2),Y
EA66 C9 7C        .|      CMP  #&7C
EA68 F0 1F        ..      BEQ  &EA89
```

```
EA6A C9 22        ."      CMP  #&22
EA6C F0 1B        ..      BEQ  &EA89
EA6E C9 21        .!      CMP  #&21
EA70 D0 05        ..      BNE  &EA77
EA72 C8           .       INY
EA73 A9 80        ..      LDA  #&80
EA75 D0 BA        ..      BNE  &EA31
EA77 C9 20        .       CMP  #&20
EA79 90 14        ..      BCC  &EA8F
EA7B C9 3F        .?      CMP  #&3F
EA7D F0 08        ..      BEQ  &EA87
EA7F 20 BF EA      ..     JSR  &EABF
EA82 2C B7 D9     ,..     BIT  &D9B7
EA85 70 03        p.      BVS  &EA8A
EA87 A9 7F        ..      LDA  #&7F
EA89 B8           .       CLV
EA8A C8           .       INY
EA8B 05 E5        ..      ORA  &E5
EA8D 18           .       CLC
EA8E 60           `       RTS
EA8F 00           .       BRK
EA90 FD 42 61     .Ba     SBC  &6142,X
EA93 64           d       ???
EA94 20 73 74      st     JSR  &7473
EA97 72           r       ???
EA98 69 6E        in      ADC  #&6E
EA9A 67           g       ???
EA9B 00           .       BRK
EA9C C9 30        .0      CMP  #&30
EA9E F0 1E        ..      BEQ  &EABE
EAA0 C9 40        .@      CMP  #&40
EAA2 F0 1A        ..      BEQ  &EABE
EAA4 90 12        ..      BCC  &EAB8
EAA6 C9 7F        ..      CMP  #&7F
EAA8 F0 14        ..      BEQ  &EABE
EAAA B0 10        ..      BCS  &EABC
EAAC 49 30        I0      EOR  #&30
EAAE C9 6F        .o      CMP  #&6F
EAB0 F0 04        ..      BEQ  &EAB6
EAB2 C9 50        .P      CMP  #&50
EAB4 D0 02        ..      BNE  &EAB8
EAB6 49 1F        I.      EOR  #&1F
EAB8 C9 21        .!      CMP  #&21
EABA 90 02        ..      BCC  &EABE
EABC 49 10        I.      EOR  #&10
EABE 60           `       RTS
EABF C9 7F        ..      CMP  #&7F
EAC1 F0 0E        ..      BEQ  &EAD1
EAC3 B0 E7        ..      BCS  &EAAC
EAC5 C9 60        .`      CMP  #&60
EAC7 D0 02        ..      BNE  &EACB
EAC9 A9 5F        ._      LDA  #&5F
EACB C9 40        .@      CMP  #&40
EACD 90 02        ..      BCC  &EAD1
EACF 29 1F        ).      AND  #&1F
EAD1 60           `       RTS
EAD2 2F           /       ???
EAD3 21 42        !B      AND  (&42,X)
EAD5 4F           O       ???
EAD6 4F           O       ???
EAD7 54           T       ???
EAD8 0D AD 87     ...     ORA  &87AD
```

'Bad string' &FD

'/!BOOT'

check for Break intercept code.

```
EADB 02          .      ???
EADC 49 4C       IL     EOR  #&4C
EADE D0 13       ..     BNE  &EAF3
EAE0 4C 87 02    L..    JMP  &0287
EAE3 AD 90 02    ...    LDA  &0290
EAE6 8E 90 02    ...    STX  &0290
EAE9 AA          .      TAX
EAEA 98          .      TYA
EAEB 29 01       ).     AND  #&01
EAED AC 91 02    ...    LDY  &0291
EAF0 8D 91 02    ...    STA  &0291
EAF3 60          `      RTS
EAF4 98          .      TYA
EAF5 9D 00 FC    ...    STA  &FC00,X
EAF8 60          `      RTS
EAF9 98          .      TYA
EAFA 9D 00 FD    ...    STA  &FD00,X
EAFD 60          `      RTS
EAFE 98          .      TYA
EAFF 9D 00 FE    ...    STA  &FE00,X
EB02 60          `      RTS
EB03 A9 04       ..     LDA  #&04
EB05 9D 08 08    ...    STA  &0808,X
EB08 A9 C0       ..     LDA  #&C0
EB0A 9D 04 08    ...    STA  &0804,X
EB0D AC 62 02    .b.    LDY  &0262
EB10 F0 02       ..     BEQ  &EB14
EB12 A9 C0       ..     LDA  #&C0
EB14 38          8      SEC
EB15 E9 40       .@     SBC  #&40
EB17 4A          J      LSR  A
EB18 4A          J      LSR  A
EB19 4A          J      LSR  A
EB1A 49 0F       I.     EOR  #&0F
EB1C 1D 3C EB    .<.    ORA  &EB3C,X
EB1F 09 10       ..     ORA  #&10
EB21 08          .      PHP
EB22 78          x      SEI
EB23 A0 FF       ..     LDY  #&FF
EB25 8C 43 FE    .C.    STY  &FE43
EB28 8D 4F FE    .O.    STA  &FE4F
EB2B C8          .      INY
EB2C 8C 40 FE    .@.    STY  &FE40
EB2F A0 02       ..     LDY  #&02
EB31 88          .      DEY
EB32 D0 FD       ..     BNE  &EB31
EB34 A0 08       ..     LDY  #&08
EB36 8C 40 FE    .@.    STY  &FE40
EB39 A0 04       ..     LDY  #&04
EB3B 88          .      DEY
EB3C D0 FD       ..     BNE  &EB3B
EB3E 28          (      PLP
EB3F 60          `      RTS
EB40 E0 C0       ..     CPX  #&C0
EB42 A0 80       ..     LDY  #&80
EB44 4C 59 EC    LY.    JMP  &EC59
EB47 A9 00       ..     LDA  #&00
EB49 8D 3B 08    .;.    STA  &083B
EB4C AD 38 08    .8.    LDA  &0838
EB4F D0 06       ..     BNE  &EB57
EB51 EE 3B 08    .;.    INC  &083B
EB54 CE 38 08    .8.    DEC  &0838
```

A= old setting *FX144 new value
X= old value.
AND new value with one to Kill(B7-B1)
Y= old value.
new value AND#1
exit.

*TV equivalent.

*FX147 Write to FRED.

*FX149 Write to JIM.

*FX151 write to SHEILA.

```
EB57 A2 08        ..      LDX  #&08
EB59 CA           .       DEX
EB5A BD 00 08     ...     LDA  &0800,X
EB5D F0 E5        ..      BEQ  &EB44
EB5F BD CF 02     ...     LDA  &02CF,X
EB62 30 05        0.      BMI  &EB69
EB64 BD 18 08     ...     LDA  &0818,X
EB67 D0 03        ..      BNE  &EB6C
EB69 20 6B EC      k.     JSR  &EC6B
EB6C BD 18 08     ...     LDA  &0818,X
EB6F F0 13        ..      BEQ  &EB84
EB71 C9 FF        ..      CMP  #&FF
EB73 F0 12        ..      BEQ  &EB87
EB75 DE 1C 08     ...     DEC  &081C,X
EB78 D0 0D        ..      BNE  &EB87
EB7A A9 05        ..      LDA  #&05
EB7C 9D 1C 08     ...     STA  &081C,X
EB7F DE 18 08     ...     DEC  &0818,X
EB82 D0 03        ..      BNE  &EB87
EB84 20 6B EC      k.     JSR  &EC6B
EB87 BD 24 08     .$.     LDA  &0824,X
EB8A F0 05        ..      BEQ  &EB91
EB8C DE 24 08     .$.     DEC  &0824,X
EB8F D0 B3        ..      BNE  &EB44
EB91 BC 20 08     . .     LDY  &0820,X
EB94 C0 FF        ..      CPY  #&FF
EB96 F0 AC        ..      BEQ  &EB44
EB98 B9 C0 08     ...     LDA  &08C0,Y
EB9B 29 7F        ).      AND  #&7F
EB9D 9D 24 08     .$.     STA  &0824,X
EBA0 BD 08 08     ...     LDA  &0808,X
EBA3 C9 04        ..      CMP  #&04
EBA5 F0 60        .`      BEQ  &EC07
EBA7 BD 08 08     ...     LDA  &0808,X
EBAA 18           .       CLC
EBAB 7D 20 08     } .     ADC  &0820,X
EBAE A8           .       TAY
EBAF B9 CB 08     ...     LDA  &08CB,Y
EBB2 38           8       SEC
EBB3 E9 3F        .?      SBC  #&3F
EBB5 8D 3A 08     .:.     STA  &083A
EBB8 B9 C7 08     ...     LDA  &08C7,Y
EBBB 8D 39 08     .9.     STA  &0839
EBBE BD 04 08     ...     LDA  &0804,X
EBC1 48           H       PHA
EBC2 18           .       CLC
EBC3 6D 39 08     m9.     ADC  &0839
EBC6 50 07        P.      BVC  &EBCF
EBC8 2A           *       ROL  A
EBC9 A9 3F        .?      LDA  #&3F
EBCB B0 02        ..      BCS  &EBCF
EBCD 49 FF        I.      EOR  #&FF
EBCF 9D 04 08     ...     STA  &0804,X
EBD2 2A           *       ROL  A
EBD3 5D 04 08     ]..     EOR  &0804,X
EBD6 10 09        ..      BPL  &EBE1
EBD8 A9 3F        .?      LDA  #&3F
EBDA 90 02        ..      BCC  &EBDE
EBDC 49 FF        I.      EOR  #&FF
EBDE 9D 04 08     ...     STA  &0804,X
EBE1 CE 39 08     .9.     DEC  &0839
EBE4 BD 04 08     ...     LDA  &0804,X
```

```
EBE7 38          8      SEC
EBE8 ED 3A 08    .:.    SBC  &083A
EBEB 4D 39 08    M9.    EOR  &0839
EBEE 30 09       0.     BMI  &EBF9
EBF0 AD 3A 08    .:.    LDA  &083A
EBF3 9D 04 08    ...    STA  &0804,X
EBF6 FE 08 08    ...    INC  &0808,X
EBF9 68          h      PLA
EBFA 5D 04 08    ]..    EOR  &0804,X
EBFD 29 F8       ).     AND  #&F8
EBFF F0 06       ..     BEQ  &EC07
EC01 BD 04 08    ...    LDA  &0804,X
EC04 20 0A EB     ..    JSR  &EB0A
EC07 BD 10 08    ...    LDA  &0810,X
EC0A C9 03       ..     CMP  #&03
EC0C F0 4B       .K     BEQ  &EC59
EC0E BD 14 08    ...    LDA  &0814,X
EC11 D0 2A       .*     BNE  &EC3D
EC13 FE 10 08    ...    INC  &0810,X
EC16 BD 10 08    ...    LDA  &0810,X
EC19 C9 03       ..     CMP  #&03
EC1B D0 10       ..     BNE  &EC2D
EC1D BC 20 08    . .    LDY  &0820,X
EC20 B9 C0 08    ...    LDA  &08C0,Y
EC23 30 34       04     BMI  &EC59
EC25 A9 00       ..     LDA  #&00
EC27 9D 30 08    .0.    STA  &0830,X
EC2A 9D 10 08    ...    STA  &0810,X
EC2D BD 10 08    ...    LDA  &0810,X
EC30 18          .      CLC
EC31 7D 20 08    } .    ADC  &0820,X
EC34 A8          .      TAY
EC35 B9 C4 08    ...    LDA  &08C4,Y
EC38 9D 14 08    ...    STA  &0814,X
EC3B F0 1C       ..     BEQ  &EC59
EC3D DE 14 08    ...    DEC  &0814,X
EC40 BD 20 08    . .    LDA  &0820,X
EC43 18          .      CLC
EC44 7D 10 08    }..    ADC  &0810,X
EC47 A8          .      TAY
EC48 B9 C1 08    ...    LDA  &08C1,Y
EC4B 18          .      CLC
EC4C 7D 30 08    }0.    ADC  &0830,X
EC4F 9D 30 08    .0.    STA  &0830,X
EC52 18          .      CLC
EC53 7D 0C 08    }..    ADC  &080C,X
EC56 20 01 ED     ..    JSR  &ED01
EC59 E0 04       ..     CPX  #&04
EC5B F0 0D       ..     BEQ  &EC6A
EC5D 4C 59 EB    LY.    JMP  &EB59
EC60 A2 08       ..     LDX  #&08
EC62 CA          .      DEX
EC63 20 A2 EC     ..    JSR  &ECA2
EC66 E0 04       ..     CPX  #&04
EC68 D0 F8       ..     BNE  &EC62
EC6A 60          `      RTS
EC6B BD 08 08    ...    LDA  &0808,X
EC6E C9 04       ..     CMP  #&04
EC70 F0 05       ..     BEQ  &EC77
EC72 A9 03       ..     LDA  #&03
EC74 9D 08 08    ...    STA  &0808,X
EC77 BD CF 02    ...    LDA  &02CF,X
```

```
EC7A F0 14        ..      BEQ  &EC90
EC7C A9 00        ..      LDA  #&00
EC7E 9D CF 02     ...     STA  &02CF,X
EC81 A0 04        ..      LDY  #&04
EC83 99 2B 08     .+.     STA  &082B,Y
EC86 88           .       DEY
EC87 D0 FA        ..      BNE  &EC83
EC89 9D 18 08     ...     STA  &0818,X
EC8C 88           .       DEY
EC8D 8C 38 08     .8.     STY  &0838
EC90 BD 28 08     .(.     LDA  &0828,X
EC93 F0 46        .F      BEQ  &ECDB
EC95 AD 3B 08     .;.     LDA  &083B
EC98 F0 36        .6      BEQ  &ECD0
EC9A A9 00        ..      LDA  #&00
EC9C 9D 28 08     .(.     STA  &0828,X
EC9F 4C 98 ED     L..     JMP  &ED98
ECA2 20 03 EB      ..     JSR  &EB03
ECA5 98           .       TYA
ECA6 9D 18 08     ...     STA  &0818,X
ECA9 9D CF 02     ...     STA  &02CF,X
ECAC 9D 00 08     ...     STA  &0800,X
ECAF A0 03        ..      LDY  #&03
ECB1 99 2C 08     .,.     STA  &082C,Y
ECB4 88           .       DEY
ECB5 10 FA        ..      BPL  &ECB1
ECB7 8C 38 08     .8.     STY  &0838
ECBA 30 4A        0J      BMI  &ED06
ECBC 08           .       PHP
ECBD 78           x       SEI
ECBE BD 08 08     ...     LDA  &0808,X
ECC1 C9 04        ..      CMP  #&04
ECC3 D0 0A        ..      BNE  &ECCF
ECC5 20 5B E4      [.     JSR  &E45B
ECC8 90 05        ..      BCC  &ECCF
ECCA A9 00        ..      LDA  #&00
ECCC 9D 00 08     ...     STA  &0800,X
ECCF 28           (       PLP
ECD0 BC 20 08     . .     LDY  &0820,X
ECD3 C0 FF        ..      CPY  #&FF
ECD5 D0 03        ..      BNE  &ECDA
ECD7 20 03 EB      ..     JSR  &EB03
ECDA 60           `       RTS
ECDB 20 5B E4      [.     JSR  &E45B
ECDE B0 DC        ..      BCS  &ECBC
ECE0 29 03        ).      AND  #&03
ECE2 F0 BB        ..      BEQ  &EC9F
ECE4 AD 38 08     .8.     LDA  &0838
ECE7 F0 15        ..      BEQ  &ECFE
ECE9 FE 28 08     .(.     INC  &0828,X
ECEC 2C 38 08     ,8.     BIT  &0838
ECEF 10 0A        ..      BPL  &ECFB
ECF1 20 5B E4      [.     JSR  &E45B
ECF4 29 03        ).      AND  #&03
ECF6 8D 38 08     .8.     STA  &0838
ECF9 10 03        ..      BPL  &ECFE
ECFB CE 38 08     .8.     DEC  &0838
ECFE 4C D0 EC     L..     JMP  &ECD0
ED01 DD 2C 08     .,.     CMP  &082C,X
ED04 F0 D4        ..      BEQ  &ECDA
ED06 9D 2C 08     .,.     STA  &082C,X
ED09 E0 04        ..      CPX  #&04
```

```
ED0B D0 09       ..      BNE  &ED16
ED0D 29 0F       ).      AND  #&0F
ED0F 1D 3C EB    .<.     ORA  &EB3C,X
ED12 08          .       PHP
ED13 4C 95 ED    L..     JMP  &ED95
ED16 48          H       PHA
ED17 29 03       ).      AND  #&03
ED19 8D 3C 08    .<.     STA  &083C
ED1C A9 00       ..      LDA  #&00
ED1E 8D 3D 08    .=.     STA  &083D
ED21 68          h       PLA
ED22 4A          J       LSR  A
ED23 4A          J       LSR  A
ED24 C9 0C       ..      CMP  #&0C
ED26 90 07       ..      BCC  &ED2F
ED28 EE 3D 08    .=.     INC  &083D
ED2B E9 0C       ..      SBC  #&0C
ED2D D0 F5       ..      BNE  &ED24
ED2F A8          .       TAY
ED30 AD 3D 08    .=.     LDA  &083D
ED33 48          H       PHA
ED34 B9 FB ED    ...     LDA  &EDFB,Y
ED37 8D 3D 08    .=.     STA  &083D
ED3A B9 07 EE    ...     LDA  &EE07,Y
ED3D 48          H       PHA
ED3E 29 03       ).      AND  #&03
ED40 8D 3E 08    .>.     STA  &083E
ED43 68          h       PLA
ED44 4A          J       LSR  A
ED45 4A          J       LSR  A
ED46 4A          J       LSR  A
ED47 4A          J       LSR  A
ED48 8D 3F 08    .?.     STA  &083F
ED4B AD 3D 08    .=.     LDA  &083D
ED4E AC 3C 08    .<.     LDY  &083C
ED51 F0 0C       ..      BEQ  &ED5F
ED53 38          8       SEC
ED54 ED 3F 08    .?.     SBC  &083F
ED57 B0 03       ..      BCS  &ED5C
ED59 CE 3E 08    .>.     DEC  &083E
ED5C 88          .       DEY
ED5D D0 F4       ..      BNE  &ED53
ED5F 8D 3D 08    .=.     STA  &083D
ED62 68          h       PLA
ED63 A8          .       TAY
ED64 F0 09       ..      BEQ  &ED6F
ED66 4E 3E 08    N>.     LSR  &083E
ED69 6E 3D 08    n=.     ROR  &083D
ED6C 88          .       DEY
ED6D D0 F7       ..      BNE  &ED66
ED6F AD 3D 08    .=.     LDA  &083D
ED72 18          .       CLC
ED73 7D 3D C4    }=.     ADC  &C43D,X
ED76 8D 3D 08    .=.     STA  &083D
ED79 90 03       ..      BCC  &ED7E
ED7B EE 3E 08    .>.     INC  &083E
ED7E 29 0F       ).      AND  #&0F
ED80 1D 3C EB    .<.     ORA  &EB3C,X
ED83 08          .       PHP
ED84 78          x       SEI
ED85 20 21 EB     !.     JSR  &EB21
ED88 AD 3D 08    .=.     LDA  &083D
```

```
ED8B 4E 3E 08     N>.     LSR  &083E
ED8E 6A           j       ROR  A
ED8F 4E 3E 08     N>.     LSR  &083E
ED92 6A           j       ROR  A
ED93 4A           J       LSR  A
ED94 4A           J       LSR  A
ED95 4C 22 EB     L".     JMP  &EB22
ED98 08           .       PHP
ED99 78           x       SEI
ED9A 20 60 E4      `.     JSR  &E460
ED9D 48           H       PHA
ED9E 29 04        ).      AND  #&04
EDA0 F0 15        ..      BEQ  &EDB7
EDA2 68           h       PLA
EDA3 BC 20 08     . .     LDY  &0820,X
EDA6 C0 FF        ..      CPY  #&FF
EDA8 D0 03        ..      BNE  &EDAD
EDAA 20 03 EB      ..     JSR  &EB03
EDAD 20 60 E4      `.     JSR  &E460
EDB0 20 60 E4      `.     JSR  &E460
EDB3 28           (       PLP
EDB4 4C F7 ED     L..     JMP  &EDF7
EDB7 68           h       PLA
EDB8 29 F8        ).      AND  #&F8
EDBA 0A           .       ASL  A
EDBB 90 0B        ..      BCC  &EDC8
EDBD 49 FF        I.      EOR  #&FF
EDBF 4A           J       LSR  A
EDC0 38           8       SEC
EDC1 E9 40        .@      SBC  #&40
EDC3 20 0A EB      ..     JSR  &EB0A
EDC6 A9 FF        ..      LDA  #&FF
EDC8 9D 20 08     . .     STA  &0820,X
EDCB A9 05        ..      LDA  #&05
EDCD 9D 1C 08     ...     STA  &081C,X
EDD0 A9 01        ..      LDA  #&01
EDD2 9D 24 08     .$.     STA  &0824,X
EDD5 A9 00        ..      LDA  #&00
EDD7 9D 14 08     ...     STA  &0814,X
EDDA 9D 08 08     ...     STA  &0808,X
EDDD 9D 30 08     .0.     STA  &0830,X
EDE0 A9 FF        ..      LDA  #&FF
EDE2 9D 10 08     ...     STA  &0810,X
EDE5 20 60 E4      `.     JSR  &E460
EDE8 9D 0C 08     ...     STA  &080C,X
EDEB 20 60 E4      `.     JSR  &E460
EDEE 28           (       PLP
EDEF 48           H       PHA
EDF0 BD 0C 08     ...     LDA  &080C,X
EDF3 20 01 ED      ..     JSR  &ED01
EDF6 68           h       PLA
EDF7 9D 18 08     ...     STA  &0818,X
EDFA 60           `       RTS
EDFB F0 B7        ..      BEQ  &EDB4
EDFD 82           .       ???
EDFE 4F           O       ???
EDFF 20 F3 C8      ..     JSR  &C8F3
EE02 A0 7B        .{      LDY  #&7B
EE04 57           W       ???
EE05 35 16        5.      AND  &16,X
EE07 E7           .       ???
EE08 D7           .       ???
```

```
EE09 CB          .      ???
EE0A C3          .      ???
EE0B B7          .      ???
EE0C AA          .      TAX
EE0D A2 9A       ..     LDX  #&9A
EE0F 92          .      ???
EE10 8A          .      TXA
EE11 82          .      ???
EE12 7A          z      ???
EE13 A9 EF       ..     LDA  #&EF
EE15 85 F5       ..     STA  &F5
EE17 60          `      RTS
EE18 A2 0D       ..     LDX  #&0D
EE1A E6 F5       ..     INC  &F5
EE1C A4 F5       ..     LDY  &F5
EE1E 10 39       .9     BPL  &EE59
EE20 A2 00       ..     LDX  #&00
EE22 86 F7       ..     STX  &F7
EE24 E8          .      INX
EE25 86 F6       ..     STX  &F6
EE27 20 BB EE     ..    JSR  &EEBB
EE2A A2 03       ..     LDX  #&03
EE2C 20 62 EE     b.    JSR  &EE62
EE2F DD 0C DF    ...    CMP  &DF0C,X
EE32 D0 E4       ..     BNE  &EE18
EE34 CA          .      DEX
EE35 10 F5       ..     BPL  &EE2C
EE37 A9 3E       .>     LDA  #&3E
EE39 85 F6       ..     STA  &F6
EE3B 20 BB EE     ..    JSR  &EEBB
EE3E A2 FF       ..     LDX  #&FF
EE40 20 62 EE     b.    JSR  &EE62
EE43 A0 08       ..     LDY  #&08
EE45 0A          .      ASL  A
EE46 76 F7       v.     ROR  &F7,X
EE48 88          .      DEY
EE49 D0 FA       ..     BNE  &EE45
EE4B E8          .      INX
EE4C F0 F2       ..     BEQ  &EE40
EE4E 18          .      CLC
EE4F 90 6A       .j     BCC  &EEBB
EE51 A2 0E       ..     LDX  #&0E
EE53 A4 F5       ..     LDY  &F5
EE55 30 0B       0.     BMI  &EE62
EE57 A0 FF       ..     LDY  #&FF
EE59 08          .      PHP
EE5A 20 68 F1     h.    JSR  &F168
EE5D 28          (      PLP
EE5E C9 01       ..     CMP  #&01
EE60 98          .      TYA
EE61 60          `      RTS
EE62 08          .      PHP
EE63 78          x      SEI
EE64 A0 10       ..     LDY  #&10
EE66 20 7F EE     ..    JSR  &EE7F
EE69 A0 00       ..     LDY  #&00
EE6B F0 17       ..     BEQ  &EE84
EE6D A0 00       ..     LDY  #&00
EE6F F0 11       ..     BEQ  &EE82
EE71 48          H      PHA
EE72 20 7A EE     z.    JSR  &EE7A
EE75 68          h      PLA
```

↑

get byte from ROM call.
current logical speech PHROM.
branch if its a PHrase ROM.
service call (get byte from ROM(X=&

write to speech processor.
Always take branch.

*FX 158 Read From Speech processor.

```
EE76 6A          j      ROR  A
EE77 6A          j      ROR  A
EE78 6A          j      ROR  A
EE79 6A          j      ROR  A
EE7A 29 0F       ).     AND  #&0F
EE7C 09 40       .@     ORA  #&40
EE7E A8          .      TAY
EE7F 98          .      TYA
EE80 A0 01       ..     LDY  #&01
EE82 08          .      PHP
EE83 78          x      SEI
EE84 2C 7B 02    ,{.    BIT  &027B
EE87 10 21       .!     BPL  &EEAA
EE89 48          H      PHA
EE8A B9 75 F0    .u.    LDA  &F075,Y
EE8D 8D 43 FE    .C.    STA  &FE43
EE90 68          h      PLA
EE91 8D 4F FE    .O.    STA  &FE4F
EE94 B9 77 F0    .w.    LDA  &F077,Y
EE97 8D 40 FE    .@.    STA  &FE40
EE9A 2C 40 FE    ,@.    BIT  &FE40
EE9D 30 FB       0.     BMI  &EE9A
EE9F AD 4F FE    .O.    LDA  &FE4F
EEA2 48          H      PHA
EEA3 B9 79 F0    .y.    LDA  &F079,Y
EEA6 8D 40 FE    .@.    STA  &FE40
EEA9 68          h      PLA
EEAA 28          (      PLP
EEAB A8          .      TAY
EEAC 60          `      RTS
EEAD AD CB 03    ...    LDA  &03CB
EEB0 85 F6       ..     STA  &F6
EEB2 AD CC 03    ...    LDA  &03CC
EEB5 85 F7       ..     STA  &F7
EEB7 A5 F5       ..     LDA  &F5
EEB9 10 1E       ..     BPL  &EED9
EEBB 08          .      PHP
EEBC 78          x      SEI
EEBD A5 F6       ..     LDA  &F6
EEBF 20 71 EE     q.    JSR  &EE71
EEC2 A5 F5       ..     LDA  &F5
EEC4 85 FA       ..     STA  &FA
EEC6 A5 F7       ..     LDA  &F7
EEC8 2A          *      ROL  A
EEC9 2A          *      ROL  A
EECA 46 FA       F.     LSR  &FA
EECC 6A          j      ROR  A
EECD 46 FA       F.     LSR  &FA
EECF 6A          j      ROR  A
EED0 20 71 EE     q.    JSR  &EE71
EED3 A5 FA       ..     LDA  &FA
EED5 20 7A EE     z.    JSR  &EE7A
EED8 28          (      PLP
EED9 60          `      RTS
EEDA A2 FF       ..     LDX  #&FF
EEDC A5 EC       ..     LDA  &EC
EEDE 05 ED       ..     ORA  &ED
EEE0 D0 06       ..     BNE  &EEE8
EEE2 A9 81       ..     LDA  #&81
EEE4 8D 4E FE    .N.    STA  &FE4E
EEE7 E8          .      INX
EEE8 8E 42 02    .B.    STX  &0242
```

*FX 159 Write to speech processor.

&F6, &F7 = 1st two spare bytes!

current ROM (for RFS) selected, branch if valid ix (0-15)

Save ST
NO IRQ's

ONLY execute if PHROM present.

```
EEEB 08          .      PHP
EEEC AD 5A 02    .Z.    LDA  &025A   read keyboard status byte
EEEF 4A          J      LSR  A       /2
EEF0 29 18       ).     AND  #&18    get bit 3&4
EEF2 09 06       ..     ORA  #&06    OR#6
EEF4 8D 40 FE    .@.    STA  &FE40   write to system VIA Reg Ø.
EEF7 4A          J      LSR  A       /2
EEF8 09 07       ..     ORA  #&07    OR#7
EEFA 8D 40 FE    .@.    STA  &FE40   write to system VIA Reg Ø.
EEFD 20 2E F1     ..    JSR  &F12E   ?&FE4Ø=8B.(%1ØØ11)
EF00 68          h      PLA          ACC=status from entry.
EF01 60          `      RTS
```

CALLED BY RESET &DA22

```
EF02 50 0A       P.     BVC  &EF0E   OSKEY
EF04 A9 01       ..     LDA  #&01    } disable CA2 interrupts
EF06 8D 4E FE    .N.    STA  &FE4E   }
EF09 B0 08       ..     BCS  &EF13
EF0B 4C 0F F0    L..    JMP  &F00F
EF0E 90 06       ..     BCC  &EF16
EF10 4C D1 F0    L..    JMP  &F0D1
EF13 EE 42 02    .B.    INC  &0242   keyboard interrupt disable flag.
EF16 AD 5A 02    .Z.    LDA  &025A   - status of KBD lights.
EF19 29 B7       ).     AND  #&B7
EF1B A2 00       ..     LDX  #&00
EF1D 20 2A F0     *.    JSR  &F02A   - test a key. (scan)
EF20 86 FA       ..     STX  &FA     - save result.
EF22 B8          .      CLV
EF23 10 05       ..     BPL  &EF2A   - branch if key isn't pressed.
EF25 2C B7 D9    ,..    BIT  &D9B7
EF28 09 08       ..     ORA  #&08    set bit 3
EF2A E8          .      INX
EF2B 20 2A F0     *.    JSR  &F02A   - test a key (scan)
EF2E 90 BB       ..     BCC  &EEEB
EF30 10 02       ..     BPL  &EF34   - branch if not pressed.
EF32 09 40       .@     ORA  #&40    - set bit 6
EF34 8D 5A 02    .Z.    STA  &025A   - keyboard status/lights
EF37 A6 EC       ..     LDX  &EC
EF39 F0 12       ..     BEQ  &EF4D
EF3B 20 2A F0     *.    JSR  &F02A
EF3E 30 10       0.     BMI  &EF50
EF40 E4 EC       ..     CPX  &EC
EF42 86 EC       ..     STX  &EC
EF44 D0 07       ..     BNE  &EF4D
EF46 A2 00       ..     LDX  #&00
EF48 86 EC       ..     STX  &EC
EF4A 20 1F F0     ..    JSR  &F01F
EF4D 4C E9 EF    L..    JMP  &EFE9
EF50 E4 EC       ..     CPX  &EC
EF52 D0 EE       ..     BNE  &EF42
EF54 A5 E7       ..     LDA  &E7
EF56 F0 23       .#     BEQ  &EF7B
EF58 C6 E7       ..     DEC  &E7
EF5A D0 1F       ..     BNE  &EF7B
EF5C AD CA 02    ...    LDA  &02CA
EF5F 85 E7       ..     STA  &E7
EF61 AD 55 02    .U.    LDA  &0255
EF64 8D CA 02    ...    STA  &02CA
EF67 AD 5A 02    .Z.    LDA  &025A
EF6A A6 EC       ..     LDX  &EC
EF6C E0 D0       ..     CPX  #&D0
EF6E D0 0E       ..     BNE  &EF7E
EF70 09 90       ..     ORA  #&90
EF72 49 A0       I.     EOR  #&A0
```

Keyboard control entry point &228,&229.

```
EF74 8D 5A 02     .Z.     STA  &025A
EF77 A9 00        ..      LDA  #&00
EF79 85 E7        ..      STA  &E7
EF7B 4C E9 EF     L..     JMP  &EFE9
EF7E E0 C0        ..      CPX  #&C0
EF80 D0 0F        ..      BNE  &EF91
EF82 09 A0        ..      ORA  #&A0
EF84 24 FA        $.      BIT  &FA
EF86 10 04        ..      BPL  &EF8C
EF88 09 10        ..      ORA  #&10
EF8A 49 80        I.      EOR  #&80
EF8C 49 90        I.      EOR  #&90
EF8E 4C 74 EF     Lt.     JMP  &EF74
EF91 BD AB EF     ...     LDA  &EFAB,X
EF94 D0 03        ..      BNE  &EF99
EF96 AD 6B 02     .k.     LDA  &026B
EF99 AE 5A 02     .Z.     LDX  &025A
EF9C 86 FA        ..      STX  &FA
EF9E 26 FA        &.      ROL  &FA
EFA0 10 07        ..      BPL  &EFA9
EFA2 A6 ED        ..      LDX  &ED
EFA4 D0 A4        ..      BNE  &EF4A
EFA6 20 BF EA      ..     JSR  &EABF
EFA9 26 FA        &.      ROL  &FA
EFAB 30 08        0.      BMI  &EFB5
EFAD 20 9C EA      ..     JSR  &EA9C
EFB0 26 FA        &.      ROL  &FA
EFB2 4C C1 EF     L..     JMP  &EFC1
EFB5 26 FA        &.      ROL  &FA
EFB7 30 0D        0.      BMI  &EFC6
EFB9 20 E3 E4      ..     JSR  &E4E3
EFBC B0 08        ..      BCS  &EFC6
EFBE 20 9C EA      ..     JSR  &EA9C
EFC1 AE 5A 02     .Z.     LDX  &025A
EFC4 10 0B        ..      BPL  &EFD1
EFC6 26 FA        &.      ROL  &FA
EFC8 10 07        ..      BPL  &EFD1
EFCA A6 ED        ..      LDX  &ED
EFCC D0 D6        ..      BNE  &EFA4
EFCE 20 9C EA      ..     JSR  &EA9C
EFD1 CD 6C 02     .l.     CMP  &026C
EFD4 D0 07        ..      BNE  &EFDD
EFD6 AE 75 02     .u.     LDX  &0275
EFD9 D0 02        ..      BNE  &EFDD
EFDB 86 E7        ..      STX  &E7
EFDD A8           .       TAY
EFDE 20 29 F1      ).     JSR  &F129
EFE1 AD 59 02     .Y.     LDA  &0259
EFE4 D0 03        ..      BNE  &EFE9
EFE6 20 F1 E4      ..     JSR  &E4F1
EFE9 A6 ED        ..      LDX  &ED
EFEB F0 0B        ..      BEQ  &EFF8
EFED 20 2A F0      *.     JSR  &F02A
EFF0 86 ED        ..      STX  &ED
EFF2 30 04        0.      BMI  &EFF8
EFF4 A2 00        ..      LDX  #&00
EFF6 86 ED        ..      STX  &ED
EFF8 A6 ED        ..      LDX  &ED
EFFA D0 16        ..      BNE  &F012
EFFC A0 EC        ..      LDY  #&EC
EFFE 20 CC F0      ..     JSR  &F0CC
F001 30 09        0.      BMI  &F00C
```

```
F003 A5 EC        ..      LDA  &EC
F005 85 ED        ..      STA  &ED
F007 86 EC        ..      STX  &EC
F009 20 1F F0      ..     JSR  &F01F
F00C 4C DA EE     L..     JMP  &EEDA
F00F 20 2A F0      *.     JSR  &F02A
F012 A5 EC        ..      LDA  &EC
F014 D0 F6        ..      BNE  &F00C
F016 A0 ED        ..      LDY  #&ED
F018 20 CC F0      ..     JSR  &F0CC
F01B 30 EF        0.      BMI  &F00C
F01D 10 E8        ..      BPL  &F007
F01F A2 01        ..      LDX  #&01
F021 86 E7        ..      STX  &E7
F023 AE 54 02     .T.     LDX  &0254
F026 8E CA 02     ...     STX  &02CA
F029 60           `       RTS
F02A A0 03        ..      LDY  #&03
F02C 8C 40 FE     .@.     STY  &FE40
F02F A0 7F        ..      LDY  #&7F
F031 8C 43 FE     .C.     STY  &FE43
F034 8E 4F FE     .O.     STX  &FE4F
F037 AE 4F FE     .O.     LDX  &FE4F
F03A 60           `       RTS
F03B 71 33        q3      ADC  (&33),Y
F03D 34           4       ???
F03E 35 84        5.      AND  &84,X
F040 38           8       SEC
F041 87           .       ???
F042 2D 5E 8C     -^.     AND  &8C5E
F045 84 EC        ..      STY  &EC
F047 86 ED        ..      STX  &ED
F049 60           `       RTS
F04A 00           .       BRK
F04B 80           .       ???
F04C 77           w       ???
F04D 65 74        et      ADC  &74
F04F 37           7       ???
F050 69 39        i9      ADC  #&39
F052 30 5F        0_      BMI  &F0B3
F054 8E 6C FE     .l.     STX  &FE6C
F057 FD 6C FA     .l.     SBC  &FA6C,X
F05A 00           .       BRK
F05B 31 32        12      AND  (&32),Y
F05D 64           d       ???
F05E 72           r       ???
F05F 36 75        6u      ROL  &75,X
F061 6F           o       ???
F062 70 5B        p[      BVS  &F0BF
F064 8F           .       ???
F065 2C B7 D9     ,..     BIT  &D9B7
F068 6C 28 02     l(.     JMP  (&0228)
F06B 01 61        .a      ORA  (&61,X)
F06D 78           x       SEI
F06E 66 79        fy      ROR  &79
F070 6A           j       ROR  A
F071 6B           k       ???
F072 40           @       RTI
F073 3A           :       ???
F074 0D 00 FF     ...     ORA  &FF00
F077 01 02        ..      ORA  (&02,X)
F079 09 0A        ..      ORA  #&0A
```

-ve inkey routine.
Acc = -ve Number EOR#&7F if pressed, Xreg -ve

*FX12Ø write current keys pressed information.

JMP(&FDFE)
JMP (&00FA) ?.

see &E9E4

```
F07B 02          .      ???
F07C 73          s      ???
F07D 63          c      ???
F07E 67          g      ???
F07F 68          h      PLA
F080 6E 6C 3B    nl;    ROR   &3B6C
F083 5D 7F AC    ]..    EOR   &AC7F,X
F086 44          D      ???
F087 02          .      ???
F088 A2 00       ..     LDX   #&00
F08A 60          `      RTS
F08B 00          .      BRK
F08C 7A          z      ???
F08D 20 76 62     vb    JSR   &6276
F090 6D 2C 2E    m,.    ADC   &2E2C
F093 2F          /      ???
F094 8B          .      ???
F095 AE 41 02    .A.    LDX   &0241
F098 4C AD E1    L..    JMP   &E1AD
F09B 1B          .      ???
F09C 81 82       ..     STA   (&82,X)
F09E 83          .      ???
F09F 85 86       ..     STA   &86
F0A1 88          .      DEY
F0A2 89          .      ???
F0A3 5C          \      ???
F0A4 8D 6C 20    .l     STA   &206C
F0A7 02          .      ???
F0A8 D0 EB       ..     BNE   &F095
F0AA A2 08       ..     LDX   #&08
F0AC 58          X      CLI
F0AD 78          x      SEI
F0AE 20 B4 F0     ..    JSR   &F0B4
F0B1 CA          .      DEX
F0B2 10 F8       ..     BPL   &F0AC
F0B4 E0 09       ..     CPX   #&09
F0B6 90 E0       ..     BCC   &F098
F0B8 60          `      RTS
F0B9 A2 09       ..     LDX   #&09
F0BB 20 68 F1     h.    JSR   &F168
F0BE 20 4A FA     J.    JSR   &FA4A
F0C1 0D 4F 53    .OS    ORA   &534F
F0C4 20 31 2E     1.    JSR   &2E31
F0C7 32          2      ???
F0C8 30 0D       0.     BMI   &F0D7
F0CA 00          .      BRK
F0CB 60          `      RTS
F0CC 18          .      CLC
F0CD A2 10       ..     LDX   #&10
F0CF B0 97       ..     BCS   &F068
F0D1 8A          .      TXA
F0D2 10 05       ..     BPL   &F0D9
F0D4 20 2A F0     *.    JSR   &F02A
F0D7 B0 55       .U     BCS   &F12E
F0D9 08          .      PHP
F0DA 90 02       ..     BCC   &F0DE
F0DC A0 EE       ..     LDY   #&EE
F0DE 99 DF 01    ...    STA   &01DF,Y
F0E1 A2 09       ..     LDX   #&09
F0E3 20 29 F1     ).    JSR   &F129
F0E6 A9 7F       ..     LDA   #&7F
F0E8 8D 43 FE    .C.    STA   &FE43
```

STY &244

*FX 131 Read top of OS RAM address (OSHWM)

JMP(8220) see &E4A1

*FX15 Flush buffer.

*FX21 Flush specific buffer.

*HELP

'OS 1.20' Not BRK! used For *HELP.

*FX 122 Keyboard scan From 16,10.

*FX 121 Keyboard Scan

```
F0EB A9 03      ..      LDA  #&03
F0ED 8D 40 FE   .@.     STA  &FE40
F0F0 A9 0F      ..      LDA  #&0F
F0F2 8D 4F FE   .O.     STA  &FE4F
F0F5 A9 01      ..      LDA  #&01
F0F7 8D 4D FE   .M.     STA  &FE4D
F0FA 8E 4F FE   .O.     STX  &FE4F
F0FD 2C 4D FE   ,M.     BIT  &FE4D
F100 F0 21      .!      BEQ  &F123
F102 8A         .       TXA
F103 D9 DF 01   ...     CMP  &01DF,Y
F106 90 16      ..      BCC  &F11E
F108 8D 4F FE   .O.     STA  &FE4F
F10B 2C 4F FE   ,O.     BIT  &FE4F
F10E 10 0E      ..      BPL  &F11E
F110 28         (       PLP
F111 08         .       PHP
F112 B0 13      ..      BCS  &F127
F114 48         H       PHA
F115 59 00 00   Y..     EOR  &0000,Y
F118 0A         .       ASL  A
F119 C9 01      ..      CMP  #&01
F11B 68         h       PLA
F11C B0 09      ..      BCS  &F127
F11E 18         .       CLC
F11F 69 10      i.      ADC  #&10
F121 10 E0      ..      BPL  &F103
F123 CA         .       DEX
F124 10 BD      ..      BPL  &F0E3
F126 8A         .       TXA
F127 AA         .       TAX
F128 28         (       PLP
F129 20 2E F1    ..     JSR  &F12E
F12C 58         X       CLI
F12D 78         x       SEI
F12E A9 0B      ..      LDA  #&0B
F130 8D 40 FE   .@.     STA  &FE40
F133 8A         .       TXA
F134 60         `       RTS
F135 49 8C      I.      EOR  #&8C
F137 0A         .       ASL  A
F138 8D 47 02   .G.     STA  &0247
F13B E0 03      ..      CPX  #&03
F13D 4C 4B F1   LK.     JMP  &F14B
F140 08         .       PHP
F141 A9 A1      ..      LDA  #&A1
F143 85 E3      ..      STA  &E3
F145 A9 19      ..      LDA  #&19
F147 8D D1 03   ...     STA  &03D1
F14A 28         (       PLP
F14B 08         .       PHP
F14C A9 06      ..      LDA  #&06
F14E 20 31 E0    1.     JSR  &E031
F151 A2 06      ..      LDX  #&06
F153 28         (       PLP
F154 F0 01      ..      BEQ  &F157
F156 CA         .       DEX
F157 86 C6      ..      STX  &C6
F159 A2 0E      ..      LDX  #&0E
F15B BD 51 D9   .Q.     LDA  &D951,X
F15E 9D 11 02   ...     STA  &0211,X
F161 CA         .       DEX
```

Reg Ø = &B (System VIA) (%ØØØØ1Ø11) &B

A=X

exit

*FX14Ø *TAPE equivalent

*FX141 *ROM equivalent

ROM/TAPE Plug (select) Ø = Tape, 1 = ROM

%10100001

set (*OPT1) to Short messages for tape & Error Abort for sequential acces

set sequential block gap. (*OPT3)

JMP (&21E) FSCV Filing System control Functions

close any open files etc. A new FS is going to take over

&C6 = 1 For 1200 baud

&C6 = 5 For (~~Rest can't~~) 300 baud

copy vectors for TFS

```
?D F162 FFFF
F162 D0 F7        ..      BNE   &F15B
F164 86 C2        ..      STX   &C2        &C2=Ø.
F166 A2 0F        ..      LDX   #&0F       All 16 ROMS ??
F168 A5 F4        ..      LDA   &F4        } save current ROM   *FX143 Issue paged ROM service request.
F16A 48           H       PHA
F16B 8A           .       TXA
F16C A2 0F        ..      LDX   #&0F
F16E FE A1 02     ...     INC   &02A1,X    } INC&DEC ROM ID table, only used to set flags!!
F171 DE A1 02     ...     DEC   &02A1,X    }
F174 10 0D        ..      BPL   &F183      skip if no service address.
F176 86 F4        ..      STX   &F4        }
F178 8E 30 FE     .0.     STX   &FE30      }
F17B 20 03 80       ..    JSR   &8003      service call
F17E AA           .       TAX              Acc holds outcome.
F17F F0 05        ..      BEQ   &F186      Ø if service has been carried out successfully.
F181 A6 F4        ..      LDX   &F4
F183 CA           .       DEX
F184 10 E8        ..      BPL   &F16E
F186 68           h       PLA              } restore previous ROM
F187 85 F4        ..      STA   &F4        }
F189 8D 30 FE     .0.     STA   &FE30      }
F18C 8A           .       TXA              A=&FF if rejected or Ø if ROM has accepted it
F18D 60           `       RTS              Return.

F18E 09 00        ..      ORA   #&00       } A<>Ø?   OSARGS entry point
F190 D0 10        ..      BNE   &F1A2      }
F192 C0 00        ..      CPY   #&00       } Y<>Ø?
F194 D0 0C        ..      BNE   &F1A2      }
F196 A5 C6        ..      LDA   &C6        A&Y must both be Ø to get here
F198 29 FB        ).      AND   #&FB       Retrieve just the baud rate bits
F19A 0D 47 02     .G.     ORA   &0247
F19D 0A           .       ASL   A
F19E 0D 47 02     .G.     ORA   &0247
F1A1 4A           J       LSR   A
F1A2 60           `       RTS

F1A3 4C F5 1D     L..     JMP   &1DF5      OSFSCV action addresses.
F1A6 F6 04        ..      INC   &04,X
F1A8 F3           .       ???
F1A9 0F           .       ???
F1AA E3           .       ???
F1AB 04           .       ???
F1AC F3           .       ???
F1AD 2A           *       ROL   A
F1AE F3           .       ???
F1AF 74           t       ???
F1B0 E2           .       ???

F1B1 C9 07        ..      CMP   #&07
F1B3 B0 ED        ..      BCS   &F1A2      Exit if call >=7   OSFSCV entry point is invalid.
F1B5 86 BC        ..      STX   &BC        temp
F1B7 0A           .       ASL   A          A*2
F1B8 AA           .       TAX              =>X
F1B9 BD A4 F1     ...     LDA   &F1A4,X    } Read Action address.
F1BC 48           H       PHA              }
F1BD BD A3 F1     ...     LDA   &F1A3,X    }
F1C0 48           H       PHA              }
F1C1 A6 BC        ..      LDX   &BC        recover from temp.
F1C3 60           `       RTS

F1C4 08           .       PHP   } same SR&A                Main load rout
F1C5 48           H       PHA   }
F1C6 20 27 FB       '.    JSR   &FB27      Set up 6850.
F1C9 AD C2 03     ...     LDA   &03C2      execution address low
```

&C6
1200=&B
300=3

```
F1CC 48          H      PHA
F1CD 20 31 F6     1.    JSR  &F631
F1D0 68          h      PLA
F1D1 F0 1A       ..     BEQ  &F1ED
F1D3 A2 03       ..     LDX  #&03
F1D5 A9 FF       ..     LDA  #&FF
F1D7 48          H      PHA
F1D8 BD BE 03    ...    LDA  &03BE,X
F1DB 95 B0       ..     STA  &B0,X
F1DD 68          h      PLA
F1DE 35 B0       5.     AND  &B0,X
F1E0 CA          .      DEX
F1E1 10 F4       ..     BPL  &F1D7
F1E3 C9 FF       ..     CMP  #&FF
F1E5 D0 06       ..     BNE  &F1ED
F1E7 20 E8 FA     ..    JSR  &FAE8
F1EA 4C 67 E2    Lg.    JMP  &E267
F1ED AD CA 03    ...    LDA  &03CA
F1F0 4A          J      LSR  A
F1F1 68          h      PLA
F1F2 F0 0E       ..     BEQ  &F202
F1F4 90 13       ..     BCC  &F209
F1F6 20 F2 FA     ..    JSR  &FAF2
F1F9 00          .      BRK
F1FA D5 4C       .L     CMP  &4C,X
F1FC 6F          o      ???
F1FD 63          c      ???
F1FE 6B          k      ???
F1FF 65 64       ed     ADC  &64
F201 00          .      BRK
F202 90 05       ..     BCC  &F209
F204 A9 03       ..     LDA  #&03
F206 8D 58 02    .X.    STA  &0258
F209 A9 30       .0     LDA  #&30
F20B 25 BB       %.     AND  &BB
F20D F0 04       ..     BEQ  &F213
F20F A5 C1       ..     LDA  &C1
F211 D0 0A       ..     BNE  &F21D
F213 98          .      TYA
F214 48          H      PHA
F215 20 BB FB     ..    JSR  &FBBB
F218 68          h      PLA
F219 A8          .      TAY
F21A 20 D5 F7     ..    JSR  &F7D5
F21D 20 B4 F9     ..    JSR  &F9B4
F220 D0 33       .3     BNE  &F255
F222 20 69 FB     i.    JSR  &FB69
F225 2C CA 03    ,..    BIT  &03CA
F228 30 08       0.     BMI  &F232
F22A 20 6A F9     j.    JSR  &F96A
F22D 20 7B F7     {.    JSR  &F77B
F230 D0 D7       ..     BNE  &F209
F232 A0 0A       ..     LDY  #&0A
F234 A5 CC       ..     LDA  &CC
F236 91 C8       ..     STA  (&C8),Y
F238 C8          .      INY
F239 A5 CD       ..     LDA  &CD
F23B 91 C8       ..     STA  (&C8),Y
F23D A9 00       ..     LDA  #&00
F23F C8          .      INY
F240 91 C8       ..     STA  (&C8),Y
F242 C8          .      INY
```

100

save it, &B4-&B7=0 & print 'Searching' on the screen

retrieve.

is it 0, branch if true

X=3

0 on Stack

retrieve

} check to see if it loads over Zero page ie Load Address = &FFFFFFFF

Branch if OK, error has occurred;

"Bad address.

block flag byte.

A=block flag byte, (off the stack)

} If A=0, branch. If bit 0 was clear then branch cassette motor off etc.

'Locked' &D5 File is locked because bit 0 of &3Cd was set (1).

branch if not 'locked'

} Break action

issue call to tube routine if present.

&BC,&BD=0, &C2=0 (or 4).

Inc block number next expected.

Branch if last block

INC &B1,&B2,&B3 by 1

Read 4 From ROM/cassette.

block header

always taken

```
F243 91 C8        ..      STA   (&C8),Y
F245 28           (       PLP
F246 20 E8 FA      ..     JSR   &FAE8
F249 24 BA        $.      BIT   &BA
F24B 30 07        O.      BMI   &F254
F24D 08           .       PHP
F24E 20 46 FA      F.     JSR   &FA46
F251 0D 00 28     ..(     ORA   &2800
F254 60           `       RTS
F255 20 37 F6      7.     JSR   &F637
F258 D0 AF        ..      BNE   &F209
F25A 86 F2        ..      STX   &F2
F25C 84 F3        ..      STY   &F3
F25E A0 00        ..      LDY   #&00
F260 20 1D EA      ..     JSR   &EA1D
F263 A2 00        ..      LDX   #&00
F265 20 2F EA      /.     JSR   &EA2F
F268 B0 0D        ..      BCS   &F277
F26A F0 08        ..      BEQ   &F274
F26C 9D D2 03     ...     STA   &03D2,X
F26F E8           .       INX
F270 E0 0B        ..      CPX   #&0B
F272 D0 F1        ..      BNE   &F265
F274 4C 8F EA     L..     JMP   &EA8F
F277 A9 00        ..      LDA   #&00
F279 9D D2 03     ...     STA   &03D2,X
F27C 60           `       RTS
F27D 48           H       PHA
F27E 86 C8        ..      STX   &C8
F280 84 C9        ..      STY   &C9
F282 A0 00        ..      LDY   #&00
F284 B1 C8        ..      LDA   (&C8),Y
F286 AA           .       TAX
F287 C8           .       INY
F288 B1 C8        ..      LDA   (&C8),Y
F28A A8           .       TAY
F28B 20 5A F2      Z.     JSR   &F25A
F28E A0 02        ..      LDY   #&02
F290 B1 C8        ..      LDA   (&C8),Y
F292 99 BC 03     ...     STA   &03BC,Y
F295 99 AE 00     ...     STA   &00AE,Y
F298 C8           .       INY
F299 C0 0A        ..      CPY   #&0A
F29B D0 F3        ..      BNE   &F290
F29D 68           h       PLA
F29E F0 07        ..      BEQ   &F2A7
F2A0 C9 FF        ..      CMP   #&FF
F2A2 D0 B0        ..      BNE   &F254
F2A4 4C C4 F1     L..     JMP   &F1C4
F2A7 8D C6 03     ...     STA   &03C6
F2AA 8D C7 03     ...     STA   &03C7
F2AD B1 C8        ..      LDA   (&C8),Y
F2AF 99 A6 00     ...     STA   &00A6,Y
F2B2 C8           .       INY
F2B3 C0 12        ..      CPY   #&12
F2B5 D0 F6        ..      BNE   &F2AD
F2B7 8A           .       TXA
F2B8 F0 BA        ..      BEQ   &F274
F2BA 20 27 FB      '.     JSR   &FB27
F2BD 20 34 F9      4.     JSR   &F934
F2C0 A9 00        ..      LDA   #&00
F2C2 20 BD FB      ..     JSR   &FBBD
```

PLP Print <CR>

save text pointer.

Validate String.

" branch if " String valid & its a null ?

branch if Bad String.

build up String in the File control block

'Bad string'

String terminator.

exit.

OSFILE entry point (A=0 for SAVE, A=FF for LOAD)

Save Acc

Save control block address.

get character from control block.

X = low address of File name
Y = high " " " "

Copy & validate File name.

Copy load address & Execution address into File control block & into Zero page.

restore Acc (indicates Save or Load).
0 = Sav
&FF = Loa

branch if Save.

is it load?

exit if not load or Save.

Jump to load routine!

SAVE

block number = 0.

Copy Start address & End address for Save into Zero page. (&B0 – &B7)

Measure length of File name from X to A.

if a null String say 'Bad string'

claim & reset 6850. For cassette use

set up 6850 & Serial ULA & cassette u

A=0

test For tube & set it up if it is explore Furth

as necessary.

```
F2C5 20 E2 FB    ..    JSR  &FBE2   Reset 6850 & set up Serial ULA.
F2C8 38          8     SEC
F2C9 A2 FD       ..    LDX  #&FD     X from &FD to &FF
F2CB BD B7 FF    ...   LDA  &FFB7,X  &B4-&B6
F2CE FD B3 FF    ...   SBC  &FFB3,X  &B0-&B2     Copy into &3C8 -&3CA
F2D1 9D CB 02    ...   STA  &02CB,X  &3C3-&3CA   (SAVE ONLY) (length of block
F2D4 E8          .     INX                                    & block Flag byte.)
F2D5 D0 F4       ..    BNE  &F2CB
F2D7 A8          .     TAY           Y= block Flag byte.
F2D8 D0 0E       ..    BNE  &F2E8    - indicates a locked File or the last block
F2DA EC C8 03    ...   CPX  &03C8    (0) CPX with length of block(low)
F2DD A9 01       ..    LDA  #&01     A=1
F2DF ED C9 03    ...   SBC  &03C9    subtract high byte block length
F2E2 90 04       ..    BCC  &F2E8    if C=0
F2E4 A2 80       ..    LDX  #&80   } always taken (indicates the last block?)
F2E6 D0 08       ..    BNE  &F2F0  }
F2E8 A9 01       ..    LDA  #&01     - block size = &100 (Default if a complete
F2EA 8D C9 03    ...   STA  &03C9     block length high                  block.)
F2ED 8E C8 03    ...   STX  &03C8     "      "      low
F2F0 8E CA 03    ...   STX  &03CA     block Flag byte.
F2F3 20 EC F7     ..   JSR  &F7EC     output the block to 6850.
F2F6 30 49       0I    BMI  &F341    - exit routine, increment a 3 byte counter (&B1,&B2,&B3).
F2F8 20 6A F9     j.   JSR  &F96A     increment block number low byte
F2FB EE C6 03    ...   INC  &03C6     loop round.
F2FE D0 C8       ..    BNE  &F2C8     block number high byte.
F300 EE C7 03    ...   INC  &03C7     always taken
F303 D0 C3       ..    BNE  &F2C8
F305 20 5A F2     Z.   JSR  &F25A     Validate File name.          OSFSC
F308 A2 FF       ..    LDX  #&FF     LSB of Execution address=&FF   A=2,
F30A 8E C2 03    ...   STX  &03C2
F30D 20 C4 F1     ..   JSR  &F1C4     the load routine.
F310 2C 7A 02    ,z.   BIT  &027A     Branch if tube not present.
F313 10 0A       ..    BPL  &F31F
F315 AD C4 03    ...   LDA  &03C4   } Are Top 2 bytes of Execution Address =&FFFF
F318 2D C5 03    -..   AND  &03C5   }
F31B C9 FF       ..    CMP  #&FF    } branch if not in I/O processor.
F31D D0 03       ..    BNE  &F322   }
F31F 6C C2 03    l..   JMP  (&03C2)  if File is to be *RUNned then call execution address
F322 A2 C2       ..    LDX  #&C2   }
F324 A0 03       ..    LDY  #&03   } Execute program in the TUBE.
F326 A9 04       ..    LDA  #&04   }
F328 4C C7 FB    L..   JMP  &FBC7   }
F32B A9 08       ..    LDA  #&08                                  OSFSC
F32D 20 44 F3     D.   JSR  &F344                                   A=5
F330 20 27 FB     '.   JSR  &FB27
F333 A9 00       ..    LDA  #&00
F335 20 48 F3     H.   JSR  &F348
F338 20 FC FA    ...   JSR  &FAFC
F33B A9 F7       ..    LDA  #&F7
F33D 25 E2       %.    AND  &E2
F33F 85 E2       ..    STA  &E2
F341 60          `     RTS
F342 A9 40       .@    LDA  #&40
F344 05 E2       ..    ORA  &E2
F346 D0 F7       ..    BNE  &F33F
F348 48          H     PHA
F349 AD 47 02    .G.   LDA  &0247   Branch if TFS selected.
F34C F0 0B       ..    BEQ  &F359
F34E 20 13 EE     ..   JSR  &EE13
F351 20 18 EE     ..   JSR  &EE18
F354 90 03       ..    BCC  &F359
F356 B8          .     CLV
```

```
F357 50 41        PA      BVC   &F39A
F359 20 7B F7      {.     JSR   &F77B
F35C AD C6 03     ...     LDA   &03C6
F35F 85 B4        ..      STA   &B4
F361 AD C7 03     ...     LDA   &03C7
F364 85 B5        ..      STA   &B5
F366 A2 FF        ..      LDX   #&FF
F368 8E DF 03     ...     STX   &03DF
F36B E8           .       INX
F36C 86 BA        ..      STX   &BA
F36E F0 06        ..      BEQ   &F376
F370 20 69 FB      i.     JSR   &FB69
F373 20 7B F7      {.     JSR   &F77B
F376 AD 47 02     .G.     LDA   &0247
F379 F0 02        ..      BEQ   &F37D
F37B 50 1D        P.      BVC   &F39A
F37D 68           h       PLA
F37E 48           H       PHA
F37F F0 2D        .-      BEQ   &F3AE
F381 20 72 FA      r.     JSR   &FA72
F384 D0 16        ..      BNE   &E39C
F386 A9 30        .0      LDA   #&30
F388 25 BB        %.      AND   &BB
F38A F0 0E        ..      BEQ   &F39A
F38C AD C6 03     ...     LDA   &03C6
F38F C5 B6        ..      CMP   &B6
F391 D0 09        ..      BNE   &F39C
F393 AD C7 03     ...     LDA   &03C7
F396 C5 B7        ..      CMP   &B7
F398 D0 02        ..      BNE   &F39C
F39A 68           h       PLA
F39B 60           `       RTS
F39C AD 47 02     .G.     LDA   &0247
F39F F0 0D        ..      BEQ   &F3AE
F3A1 20 AD EE      ..     JSR   &EEAD
F3A4 A9 FF        ..      LDA   #&FF
F3A6 8D C6 03     ...     STA   &03C6
F3A9 8D C7 03     ...     STA   &03C7
F3AC D0 C2        ..      BNE   &F370
F3AE 50 05        P.      BVC   &F3B5
F3B0 A9 FF        ..      LDA   #&FF
F3B2 20 D7 F7      ..     JSR   &F7D7
F3B5 A2 00        ..      LDX   #&00
F3B7 20 D9 F9      ..     JSR   &F9D9
F3BA AD 47 02     .G.     LDA   &0247
F3BD F0 04        ..      BEQ   &F3C3
F3BF 24 BB        $.      BIT   &BB
F3C1 50 DE        P.      BVC   &F3A1
F3C3 2C CA 03     ,..     BIT   &03CA
F3C6 30 DC        0.      BMI   &F3A4
F3C8 10 A6        ..      BPL   &F370
F3CA 85 BC        ..      STA   &BC
F3CC 8A           .       TXA
F3CD 48           H       PHA
F3CE 98           .       TYA
F3CF 48           H       PHA
F3D0 A5 BC        ..      LDA   &BC
F3D2 D0 1E        ..      BNE   &F3F2
F3D4 98           .       TYA
F3D5 D0 0C        ..      BNE   &F3E3
F3D7 20 75 E2      u.     JSR   &E275
F3DA 20 78 F4      x.     JSR   &F478
```

Read the next Block Header (Type)

Copy block number into &B4, &B5.

&3DF (block flag byte copy) = &FF

&BA = 0

always taken if LDX#&FF is executed.

copy current block number read to &B4, &B5.

Read the next block header.

Branch if TFS selected.

RFS ONLY

Recover A (&FF for load)

branch if Not loading (Saving)?

check if correct file is being loaded (Filen chec

Branch if *OPT 2,0 used.

check if correct Block has been loaded, branch if not.

Recover A then Exit

Branch if TFS selected

Manipulate data in the PHROM (if present

block number = &FF,&FF

always taken.

&BD = &FF, &BC = 0, &C2 = 4 if block not empty ELSE =

branch if TFS selected.

RFS only.

branch if last block

branch if not last block.

OSFIND entry point.

```
F3DD 46 E2        F.      LSR  &E2
F3DF 06 E2        ..      ASL  &E2
F3E1 90 0C        ..      BCC  &F3EF
F3E3 4A           J       LSR  A
F3E4 B0 F7        ..      BCS  &F3DD
F3E6 4A           J       LSR  A
F3E7 B0 03        ..      BCS  &F3EC
F3E9 4C B1 FB     L..     JMP  &FBB1
F3EC 20 78 F4      x.     JSR  &F478
F3EF 4C 71 F4     Lq.     JMP  &F471
F3F2 20 5A F2      Z.     JSR  &F25A
F3F5 24 BC        $.      BIT  &BC
F3F7 50 3D        P=      BVC  &F436
F3F9 A9 00        ..      LDA  #&00
F3FB 8D 9E 03     ...     STA  &039E
F3FE 8D DD 03     ...     STA  &03DD
F401 8D DE 03     ...     STA  &03DE
F404 A9 3E        .>      LDA  #&3E
F406 20 3D F3      =.     JSR  &F33D
F409 20 1A FB      ..     JSR  &FB1A
F40C 08           .       PHP
F40D 20 31 F6      1.     JSR  &F631
F410 20 B4 F6      ..     JSR  &F6B4
F413 28           (       PLP
F414 A2 FF        ..      LDX  #&FF
F416 E8           .       INX
F417 BD B2 03     ...     LDA  &03B2,X
F41A 9D A7 03     ...     STA  &03A7,X
F41D D0 F7        ..      BNE  &F416
F41F A9 01        ..      LDA  #&01
F421 20 44 F3      D.     JSR  &F344
F424 AD EA 02     ...     LDA  &02EA
F427 0D EB 02     ...     ORA  &02EB
F42A D0 03        ..      BNE  &F42F
F42C 20 42 F3      B.     JSR  &F342
F42F A9 01        ..      LDA  #&01
F431 0D 47 02     .G.     ORA  &0247
F434 D0 39        .9      BNE  &F46F
F436 8A           .       TXA
F437 D0 03        ..      BNE  &F43C
F439 4C 8F EA     L..     JMP  &EA8F
F43C A2 FF        ..      LDX  #&FF
F43E E8           .       INX
F43F BD D2 03     ...     LDA  &03D2,X
F442 9D 80 03     ...     STA  &0380,X
F445 D0 F7        ..      BNE  &F43E
F447 A9 FF        ..      LDA  #&FF
F449 A2 08        ..      LDX  #&08
F44B 9D 8B 03     ...     STA  &038B,X
F44E CA           .       DEX
F44F D0 FA        ..      BNE  &F44B
F451 8A           .       TXA
F452 A2 14        ..      LDX  #&14
F454 9D 80 03     ...     STA  &0380,X
F457 E8           .       INX
F458 E0 1E        ..      CPX  #&1E
F45A D0 F8        ..      BNE  &F454
F45C 2E 97 03     ...     ROL  &0397
F45F 20 27 FB      '.     JSR  &FB27
F462 20 34 F9      4.     JSR  &F934
F465 20 F2 FA      ..     JSR  &FAF2
F468 A9 02        ..      LDA  #&02
```

```
F46A 20 44 F3      D.     JSR  &F344
F46D A9 02         ..     LDA  #&02
F46F 85 BC         ..     STA  &BC
F471 68            h      PLA
F472 A8            .      TAY
F473 68            h      PLA
F474 AA            .      TAX
F475 A5 BC         ..     LDA  &BC
F477 60            `      RTS
F478 A9 02         ..     LDA  #&02
F47A 25 E2         %.     AND  &E2
F47C F0 F9         ..     BEQ  &F477
F47E A9 00         ..     LDA  #&00
F480 8D 97 03      ...    STA  &0397
F483 A9 80         ..     LDA  #&80
F485 AE 9D 03      ...    LDX  &039D
F488 8E 96 03      ...    STX  &0396
F48B 8D 98 03      ...    STA  &0398
F48E 20 96 F4       ..    JSR  &F496
F491 A9 FD         ..     LDA  #&FD
F493 4C 3D F3      L=.    JMP  &F33D
F496 20 1A FB       ..    JSR  &FB1A
F499 A2 11         ..     LDX  #&11
F49B BD 8C 03      ...    LDA  &038C,X
F49E 9D BE 03      ...    STA  &03BE,X
F4A1 CA            .      DEX
F4A2 10 F7         ..     BPL  &F49B
F4A4 86 B2         ..     STX  &B2
F4A6 86 B3         ..     STX  &B3
F4A8 E8            .      INX
F4A9 86 B0         ..     STX  &B0
F4AB A9 09         ..     LDA  #&09
F4AD 85 B1         ..     STA  &B1
F4AF A2 7F         ..     LDX  #&7F
F4B1 20 81 FB       ..    JSR  &FB81
F4B4 8D DF 03      ...    STA  &03DF
F4B7 20 8E FB       ..    JSR  &FB8E
F4BA 20 E2 FB       ..    JSR  &FBE2
F4BD 20 EC F7       ..    JSR  &F7EC
F4C0 EE 94 03      ...    INC  &0394
F4C3 D0 03         ..     BNE  &F4C8
F4C5 EE 95 03      ...    INC  &0395
F4C8 60            `      RTS
F4C9 8A            .      TXA
F4CA 48            H      PHA
F4CB 98            .      TYA
F4CC 48            H      PHA
F4CD A9 01         ..     LDA  #&01
F4CF 20 9C FB       ..    JSR  &FB9C
F4D2 A5 E2         ..     LDA  &E2
F4D4 0A            .      ASL  A
F4D5 B0 4C         .L     BCS  &F523
F4D7 0A            .      ASL  A
F4D8 90 09         ..     BCC  &F4E3
F4DA A9 80         ..     LDA  #&80
F4DC 20 44 F3      D.     JSR  &F344
F4DF A9 FE         ..     LDA  #&FE
F4E1 B0 38         .8     BCS  &F51B
F4E3 AE 9E 03      ...    LDX  &039E
F4E6 E8            .      INX
F4E7 EC EA 02      ...    CPX  &02EA
F4EA D0 2A         .*     BNE  &F516
```

OSBGET entry point

```
F4EC 2C EC 02     ,..     BIT  &02EC
F4EF 30 22        0"      BMI  &F513
F4F1 AD ED 02     ...     LDA  &02ED
F4F4 48           H       PHA
F4F5 20 1A FB      ..     JSR  &FB1A
F4F8 08           .       PHP
F4F9 20 AC F6      ..     JSR  &F6AC
F4FC 28           (       PLP
F4FD 68           h       PLA
F4FE 85 BC        ..      STA  &BC
F500 18           .       CLC
F501 2C EC 02     ,..     BIT  &02EC
F504 10 17        ..      BPL  &F51D
F506 AD EA 02     ...     LDA  &02EA
F509 0D EB 02     ...     ORA  &02EB
F50C D0 0F        ..      BNE  &F51D
F50E 20 42 F3      B.     JSR  &F342
F511 D0 0A        ..      BNE  &F51D
F513 20 42 F3      B.     JSR  &F342
F516 CA           .       DEX
F517 18           .       CLC
F518 BD 00 0A     ...     LDA  &0A00,X
F51B 85 BC        ..      STA  &BC
F51D EE 9E 03     ...     INC  &039E
F520 4C 71 F4     Lq.     JMP  &F471
F523 00           .       BRK
F524 DF           .       ???
F525 45 4F        EO      EOR  &4F
F527 46 00        F.      LSR  &00
F529 85 C4        ..      STA  &C4
F52B 8A           .       TXA
F52C 48           H       PHA
F52D 98           .       TYA
F52E 48           H       PHA
F52F A9 02        ..      LDA  #&02
F531 20 9C FB      ..     JSR  &FB9C
F534 AE 9D 03     ...     LDX  &039D
F537 A5 C4        ..      LDA  &C4
F539 9D 00 09     ...     STA  &0900,X
F53C E8           .       INX
F53D D0 06        ..      BNE  &F545
F53F 20 96 F4      ..     JSR  &F496
F542 20 F2 FA      ..     JSR  &FAF2
F545 EE 9D 03     ...     INC  &039D
F548 A5 C4        ..      LDA  &C4
F54A 4C 6F F4     Lo.     JMP  &F46F
F54D 8A           .       TXA
F54E F0 2E        ..      BEQ  &F57E
F550 E0 03        ..      CPX  #&03
F552 F0 1F        ..      BEQ  &F573
F554 C0 03        ..      CPY  #&03
F556 B0 06        ..      BCS  &F55E
F558 CA           .       DEX
F559 F0 06        ..      BEQ  &F561
F55B CA           .       DEX
F55C F0 0A        ..      BEQ  &F568
F55E 4C 10 E3     L..     JMP  &E310
F561 A9 33        .3      LDA  #&33
F563 C8           .       INY
F564 C8           .       INY
F565 C8           .       INY
F566 D0 02        ..      BNE  &F56A
```

'EOF' &DF

OSBPUT entry point

OSFSCV A=0.

```
F568 A9 CC        ..      LDA   #&CC
F56A C8           .       INY
F56B 25 E3        %.      AND   &E3
F56D 19 81 F5     ...     ORA   &F581,Y
F570 85 E3        ..      STA   &E3
F572 60           `       RTS
F573 98           .       TYA
F574 30 02        0.      BMI   &F578
F576 D0 02        ..      BNE   &F57A
F578 A9 19        ..      LDA   #&19
F57A 8D D1 03     ...     STA   &03D1
F57D 60           `       RTS
F57E A8           .       TAY
F57F F0 EC        ..      BEQ   &F56D
F581 A1 00        ..      LDA   (&00,X)
F583 22           "       ???
F584 11 00        ..      ORA   (&00),Y
F586 88           .       DEY
F587 CC C6 C0     ...     CPY   &C0C6
F58A AD 47 02     .G.     LDA   &0247
F58D F0 07        ..      BEQ   &F596
F58F 20 51 EE      Q.     JSR   &EE51
F592 A8           .       TAY
F593 18           .       CLC
F594 90 1A        ..      BCC   &F5B0
F596 AD 08 FE     ...     LDA   &FE08
F599 48           H       PHA
F59A 29 02        ).      AND   #&02
F59C F0 0B        ..      BEQ   &F5A9
F59E A4 CA        ..      LDY   &CA
F5A0 F0 07        ..      BEQ   &F5A9
F5A2 68           h       PLA
F5A3 A5 BD        ..      LDA   &BD
F5A5 8D 09 FE     ...     STA   &FE09
F5A8 60           `       RTS
F5A9 AC 09 FE     ...     LDY   &FE09
F5AC 68           h       PLA
F5AD 4A           J       LSR   A
F5AE 4A           J       LSR   A
F5AF 4A           J       LSR   A
F5B0 A6 C2        ..      LDX   &C2
F5B2 F0 69        .i      BEQ   &F61D
F5B4 CA           .       DEX
F5B5 D0 06        ..      BNE   &F5BD
F5B7 90 64        .d      BCC   &F61D
F5B9 A0 02        ..      LDY   #&02
F5BB D0 5E        .^      BNE   &F61B
F5BD CA           .       DEX
F5BE D0 13        ..      BNE   &F5D3
F5C0 B0 5B        .[      BCS   &F61D
F5C2 98           .       TYA
F5C3 20 78 FB      x.     JSR   &FB78
F5C6 A0 03        ..      LDY   #&03
F5C8 C9 2A        .*      CMP   #&2A
F5CA F0 4F        .O      BEQ   &F61B
F5CC 20 50 FB      P.     JSR   &FB50
F5CF A0 01        ..      LDY   #&01
F5D1 D0 48        .H      BNE   &F61B
F5D3 CA           .       DEX
F5D4 D0 0C        ..      BNE   &F5E2
F5D6 B0 04        ..      BCS   &F5DC
F5D8 84 BD        ..      STY   &BD
```

See &DCB0.

CPY #&AD

cassette/ROM Flag.

branch if tape.

get byte from ROM. (in the IRQ?)

always taken

6850 - Status register.

save ST (6850) on Stack

Branch if TX register not empty (ie when Reading)

If &CA=0 then read when TX IRQ occurs (?)

remove from Stack (6850 status)

read byte being saved from &BD.

data reg (TX) - 6850 (send to 6850)

data reg (RX) - 6850.

retrieve 6850 status.

C = B2 of 6850 status (carrier present 1= no, 0=yes)

Exit routine if &C2=0.

&C2=1

exit if B2=0 (carrier is present on cassette input)

Always taken.

&C2=2

exit if B2=1 (carrier not present)

A = byte from 6850 (data)

&BE, &BF, &C0 = 0

Branch if Sync byte found.

Initialise serial ULA &6850.

Y=1 and branch.

&C2=3

branch if carrier not present.

Save byte read from 6850.

data

```
F5DA F0 41        .A    BEQ  &F61D
F5DC A9 80        ..    LDA  #&80
F5DE 85 C0        ..    STA  &C0
F5E0 D0 3B        .;    BNE  &F61D
F5E2 CA           .     DEX
F5E3 D0 29        .)    BNE  &F60E
F5E5 B0 2F        ./    BCS  &F616
F5E7 98           .     TYA
F5E8 20 B0 F7      ..   JSR  &F7B0
F5EB A4 BC        ..    LDY  &BC
F5ED E6 BC        ..    INC  &BC
F5EF 24 BD        $.    BIT  &BD
F5F1 30 0D        0.    BMI  &F600
F5F3 20 D3 FB      ..   JSR  &FBD3
F5F6 F0 05        ..    BEQ  &F5FD
F5F8 8E E5 FE     ...   STX  &FEE5
F5FB D0 03        ..    BNE  &F600
F5FD 8A           .     TXA
F5FE 91 B0        ..    STA  (&B0),Y
F600 C8           .     INY
F601 CC C8 03     ...   CPY  &03C8
F604 D0 17        ..    BNE  &F61D
F606 A9 01        ..    LDA  #&01
F608 85 BC        ..    STA  &BC
F60A A0 05        ..    LDY  #&05
F60C D0 0D        ..    BNE  &F61B
F60E 98           .     TYA
F60F 20 B0 F7      ..   JSR  &F7B0
F612 C6 BC        ..    DEC  &BC
F614 10 07        ..    BPL  &F61D
F616 20 46 FB      F.   JSR  &FB46
F619 A0 00        ..    LDY  #&00
F61B 84 C2        ..    STY  &C2
F61D 60           `     RTS
F61E 48           H     PHA
F61F 98           .     TYA
F620 48           H     PHA
F621 8A           .     TXA
F622 A8           .     TAY
F623 A9 03        ..    LDA  #&03
F625 20 9C FB      ..   JSR  &FB9C
F628 A5 E2        ..    LDA  &E2
F62A 29 40        )@    AND  #&40
F62C AA           .     TAX
F62D 68           h     PLA
F62E A8           .     TAY
F62F 68           h     PLA
F630 60           `     RTS
F631 A9 00        ..    LDA  #&00
F633 85 B4        ..    STA  &B4
F635 85 B5        ..    STA  &B5
F637 A5 B4        ..    LDA  &B4
F639 48           H     PHA
F63A 85 B6        ..    STA  &B6
F63C A5 B5        ..    LDA  &B5
F63E 48           H     PHA
F63F 85 B7        ..    STA  &B7
F641 20 46 FA      F.   JSR  &FA46
F644 53           S     ???
F645 65 61        ea    ADC  &61
F647 72           r     ???
F648 63           c     ???
```

always taken (exit)

branch if corner not present.
A = data byte from 6850,
perform CRC
Read and check CRC result (??)

check if tube present & whether loading in I/O processor; branch if Not For TUBE.
Send across TUBE
always taken.

length of block (low)
exit if not equal
&BC = 1
? & C2 = 5
A = data byte from 6850.
Perform CRC
Check CRC result (?)
reset 6850

&C2 = 3 if sync has just been found
&C2 = 2 if corner not present
&C2 = 1 if byte other than sync in

OSFSC(8) A = 1

block high & low = 0.

&B4 - &B7 = 0.

0 on stack

0 on stack

?.

('Searching' Not error message.)

```
F649 68          h      PLA
F64A 69 6E       in     ADC  #&6E
F64C 67          g      ???
F64D 0D 00 A9    ...    ORA  &A900
F650 FF          .      ???
F651 20 48 F3     H.    JSR  &F348
F654 68          h      PLA
F655 85 B5       ..     STA  &B5
F657 68          h      PLA
F658 85 B4       ..     STA  &B4
F65A A5 B6       ..     LDA  &B6
F65C 05 B7       ..     ORA  &B7
F65E D0 0D       ..     BNE  &F66D
F660 85 B4       ..     STA  &B4
F662 85 B5       ..     STA  &B5
F664 A5 C1       ..     LDA  &C1
F666 D0 05       ..     BNE  &F66D
F668 A2 B1       ..     LDX  #&B1
F66A 20 81 FB     ..    JSR  &FB81
F66D AD 47 02    .G.    LDA  &0247
F670 F0 13       ..     BEQ  &F685
F672 70 11       p.     BVS  &F685
F674 00          .      BRK
F675 D6 46       .F     DEC  &46,X
F677 69 6C       il     ADC  #&6C
F679 65 20       e      ADC  &20
F67B 6E 6F 74    not    ROR  &746F
F67E 20 66 6F     fo    JSR  &6F66
F681 75 6E       un     ADC  &6E,X
F683 64          d      ???
F684 00          .      BRK
F685 A0 FF       ..     LDY  #&FF
F687 8C DF 03    ...    STY  &03DF
F68A 60          `      RTS
F68B A9 00       ..     LDA  #&00
F68D 08          .      PHP
F68E 84 E6       ..     STY  &E6
F690 AC 56 02    .V.    LDY  &0256
F693 8D 56 02    .V.    STA  &0256
F696 F0 03       ..     BEQ  &F69B
F698 20 CE FF     ..    JSR  OSFIND
F69B A4 E6       ..     LDY  &E6
F69D 28          (      PLP
F69E F0 0B       ..     BEQ  &F6AB
F6A0 A9 40       .@     LDA  #&40
F6A2 20 CE FF     ..    JSR  OSFIND
F6A5 A8          .      TAY
F6A6 F0 CC       ..     BEQ  &F674
F6A8 8D 56 02    .V.    STA  &0256
F6AB 60          `      RTS
F6AC A2 A6       ..     LDX  #&A6
F6AE 20 81 FB     ..    JSR  &FB81
F6B1 20 7B F7     {.    JSR  &F77B
F6B4 AD CA 03    ...    LDA  &03CA
F6B7 4A          J      LSR  A
F6B8 90 03       ..     BCC  &F6BD
F6BA 4C F6 F1    L..    JMP  &F1F6
F6BD AD DD 03    ...    LDA  &03DD
F6C0 85 B4       ..     STA  &B4
F6C2 AD DE 03    ...    LDA  &03DE
F6C5 85 B5       ..     STA  &B5
F6C7 A9 00       ..     LDA  #&00
```

~~A9~~ LDA #&FF

'File not Found' &D6

*EXEC

```
F6C9 85 B0        ..      STA   &B0
F6CB A9 0A        ..      LDA   #&0A
F6CD 85 B1        ..      STA   &B1
F6CF A9 FF        ..      LDA   #&FF
F6D1 85 B2        ..      STA   &B2
F6D3 85 B3        ..      STA   &B3
F6D5 20 D5 F7      ..     JSR   &F7D5
F6D8 20 B4 F9      ..     JSR   &F9B4
F6DB D0 25        .%      BNE   &F702
F6DD AD FF 0A     ...     LDA   &0AFF
F6E0 8D ED 02     ...     STA   &02ED
F6E3 20 69 FB      i.     JSR   &FB69
F6E6 8E DD 03     ...     STX   &03DD
F6E9 8C DE 03     ...     STY   &03DE
F6EC A2 02        ..      LDX   #&02
F6EE BD C8 03     ...     LDA   &03C8,X
F6F1 9D EA 02     ...     STA   &02EA,X
F6F4 CA           .       DEX
F6F5 10 F7        ..      BPL   &F6EE
F6F7 2C EC 02     ,..     BIT   &02EC
F6FA 10 03        ..      BPL   &F6FF
F6FC 20 49 F2      I.     JSR   &F249
F6FF 4C F2 FA     L..     JMP   &FAF2
F702 20 37 F6      7.     JSR   &F637
F705 D0 AD        ..      BNE   &F6B4
F707 C9 2A        .*      CMP   #&2A
F709 F0 37        .7      BEQ   &F742
F70B C9 23        .#      CMP   #&23
F70D D0 0F        ..      BNE   &F71E
F70F EE C6 03     ...     INC   &03C6
F712 D0 03        ..      BNE   &F717
F714 EE C7 03     ...     INC   &03C7
F717 A2 FF        ..      LDX   #&FF
F719 2C B7 D9     ,..     BIT   &D9B7
F71C D0 55        .U      BNE   &F773
F71E A9 F7        ..      LDA   #&F7
F720 20 3D F3      =.     JSR   &F33D
F723 00           .       BRK
F724 D7           .       ???
F725 42           B       ???
F726 61 64        ad      ADC   (&64,X)
F728 20 52 4F      RO     JSR   &4F52
F72B 4D 00 A0     M..     EOR   &A000
F72E FF           .       ???
F72F 20 90 FB      ..     JSR   &FB90
F732 A9 01        ..      LDA   #&01
F734 85 C2        ..      STA   &C2
F736 20 50 FB      P.     JSR   &FB50
F739 20 95 F9      ..     JSR   &F995
F73C A9 03        ..      LDA   #&03
F73E C5 C2        ..      CMP   &C2
F740 D0 F7        ..      BNE   &F739
F742 A0 00        ..      LDY   #&00
F744 20 7C FB      !.     JSR   &FB7C
F747 20 97 F7      ..     JSR   &F797
F74A 50 1A        P.      BVC   &F766
F74C 99 B2 03     ...     STA   &03B2,Y
F74F F0 06        ..      BEQ   &F757
F751 C8           .       INY
F752 C0 0B        ..      CPY   #&0B
F754 D0 F1        ..      BNE   &F747
F756 88           .       DEY
```

'Bad ROM' &D7

Read Block Header

LDY #&FF

? &C3=&FF (Y reg) & turn the relay on.

&C2=1

Reset 6850 & initialise the Serial ULA

test escape (also looks at cassette cutout flag but this seems irrele

wait until? &C2=3

Y=0, ? &BE=0 ?, &BF=0 (From Y reg)

get byte From ROM/Cassette & update CRC.

bit 6 of &C0.

Read 11 bytes from cassette/ROM to &3B2+&3BC

Y=&A.

```
F757 A2 0C        ..      LDX  #&0C      X=&C, Y=&A
F759 20 97 F7      ..     JSR  &F797     Read another byte
F75C 50 08        P.      BVC  &F766
F75E 9D B2 03     ...     STA  &03B2,X   Read 19 bytes into &3BE to &3D0
F761 E8           .       INX            (File Attributes).
F762 E0 1F        ..      CPX  #&1F
F764 D0 F3        ..      BNE  &F759
F766 98           .       TYA            A= &A
F767 AA           .       TAX            X= &A
F768 A9 00        ..      LDA  #&00
F76A 99 B2 03     ...     STA  &03B2,Y   ? &3BC=0
F76D A5 BE        ..      LDA  &BE
F76F 05 BF        ..      ORA  &BF
F771 85 C1        ..      STA  &C1
F773 20 78 FB      x.     JSR  &FB78     &C0=0, &BE=0, &BF=0.
F776 84 C2        ..      STY  &C2       &C2=0 (Y=0).
F778 8A           .       TXA            } A=&A, Branch always taken. (RTS)
F779 D0 59        .Y      BNE  &F7D4     }
F77B AD 47 02     .G.     LDA  &0247     Branch if TAPE filing system active.
F77E F0 AD        ..      BEQ  &F72D
F780 20 51 EE      Q.     JSR  &EE51
F783 C9 2B        .+      CMP  #&2B
F785 D0 80        ..      BNE  &F707
F787 A9 08        ..      LDA  #&08
F789 25 E2        %.      AND  &E2
F78B F0 03        ..      BEQ  &F790
F78D 20 4D F2      M.     JSR  &F24D
F790 20 18 EE      ..     JSR  &EE18
F793 90 EB        ..      BCC  &F780
F795 B8           .       CLV
F796 60           `       RTS
F797 AD 47 02     .G.     LDA  &0247     Branch if TAPE filing system active.
F79A F0 11        ..      BEQ  &F7AD
F79C 8A           .       TXA
F79D 48           H       PHA            Save X and Y on stack.
F79E 98           .       TYA
F79F 48           H       PHA
F7A0 20 51 EE      Q.     JSR  &EE51
F7A3 85 BD        ..      STA  &BD
F7A5 A9 FF        ..      LDA  #&FF      ? &C0=&FF
F7A7 85 C0        ..      STA  &C0
F7A9 68           h       PLA
F7AA A8           .       TAY            Recover X and Y from Stack.
F7AB 68           h       PLA
F7AC AA           .       TAX
F7AD 20 84 F8      ..     JSR  &F884     Get byte from cassette into &BD & A.
F7B0 08           .       PHP            } Save SR & A.
F7B1 48           H       PHA            }
F7B2 38           8       SEC            C=1
F7B3 66 CB        f.      ROR  &CB       C -> &CB
F7B5 45 BF        E.      EOR  &BF       |
F7B7 85 BF        ..      STA  &BF       |
F7B9 A5 BF        ..  ->  LDA  &BF       |
F7BB 2A           *       ROL  A         <- C
F7BC 90 0C        ..      BCC  &F7CA     |
F7BE 6A           j       ROR  A         |
F7BF 49 08        I.      EOR  #&08      |
F7C1 85 BF        ..      STA  &BF       |
F7C3 A5 BE        ..      LDA  &BE       |
F7C5 49 10        I.      EOR  #&10      |
F7C7 85 BE        ..      STA  &BE       |
F7C9 38           8       SEC            |
```

CRC calculator

P 348
AMG

```
F7CA 26 BE        &.    ROL   &BE
F7CC 26 BF        &.    ROL   &BF
F7CE 46 CB        F.    LSR   &CB
F7D0 D0 E7        ..    BNE   &F7B9
F7D2 68           h     PLA
F7D3 28           (     PLP
F7D4 60           `     RTS
F7D5 A9 00        ..    LDA   #&00
F7D7 85 BD        ..    STA   &BD
F7D9 A2 00        ..    LDX   #&00
F7DB 86 BC        ..    STX   &BC
F7DD 50 0A        P.    BVC   &F7E9
F7DF AD C8 03     ...   LDA   &03C8
F7E2 0D C9 03     ...   ORA   &03C9
F7E5 F0 02        ..    BEQ   &F7E9
F7E7 A2 04        ..    LDX   #&04
F7E9 86 C2        ..    STX   &C2
F7EB 60           `     RTS
F7EC 08           .     PHP
F7ED A2 03        ..    LDX   #&03
F7EF A9 00        ..    LDA   #&00
F7F1 9D CB 03     ...   STA   &03CB,X
F7F4 CA           .     DEX
F7F5 10 FA        ..    BPL   &F7F1
F7F7 AD C6 03     ...   LDA   &03C6
F7FA 0D C7 03     ...   ORA   &03C7
F7FD D0 05        ..    BNE   &F804
F7FF 20 92 F8      ..   JSR   &F892
F802 F0 03        ..    BEQ   &F807
F804 20 96 F8      ..   JSR   &F896
F807 A9 2A        .*    LDA   #&2A
F809 85 BD        ..    STA   &BD
F80B 20 78 FB      x.   JSR   &FB78
F80E 20 4A FB      J.   JSR   &FB4A
F811 20 84 F8      ..   JSR   &F884
F814 88           .     DEY
F815 C8           .     INY
F816 B9 D2 03     ...   LDA   &03D2,Y
F819 99 B2 03     ...   STA   &03B2,Y
F81C 20 75 F8      u.   JSR   &F875
F81F D0 F4        ..    BNE   &F815
F821 A2 0C        ..    LDX   #&0C
F823 BD B2 03     ...   LDA   &03B2,X
F826 20 75 F8      u.   JSR   &F875
F829 E8           .     INX
F82A E0 1D        ..    CPX   #&1D
F82C D0 F5        ..    BNE   &F823
F82E 20 7B F8      {.   JSR   &F87B
F831 AD C8 03     ...   LDA   &03C8
F834 0D C9 03     ...   ORA   &03C9
F837 F0 1C        ..    BEQ   &F855
F839 A0 00        ..    LDY   #&00
F83B 20 7C FB      |.   JSR   &FB7C
F83E B1 B0        ..    LDA   (&B0),Y
F840 20 D3 FB      ..   JSR   &FBD3
F843 F0 03        ..    BEQ   &F848
F845 AE E5 FE     ...   LDX   &FEE5
F848 8A           .     TXA
F849 20 75 F8      u.   JSR   &F875
F84C C8           .     INY
F84D CC C8 03     ...   CPY   &03C8
F850 D0 EC        ..    BNE   &F83E
```

&BC, &BD = Ø.

(if V=Ø & block length = Ø)

&C2 = Ø or 4

save status.

4 spare bytes, all zero.

output a block of data to ta

block number (low)
block number (high).
branch if block number <> Ø.
delay (using CFS timeout counter) (long delay
always taken after JSR &F892
delay (using inter-block gap)

SAVE the Sync byte in &BD for IRQ routi

&BE, &BF, &C0 = Ø
&C1 = &30 then alter 6850 control register.
wait until Sync has been sent to 6250.

at this point Y is Ø (on entry).

File name of program being saved
File name
Calculate CRC (for file name)
until Ø-terminator of file name

Calculate CRC — for load address etc etc
P279 AUG
Four spare byte Inte

Save CRCs to tape
length of block (low)
length of block (high)
branch if empty block.
Offset of byte being saved = Ø.
&BE &BF = Y
read byte to be saved.
~~x = byte being saved. save byte.~~
branch if no tube connected.
X = byte from 2nd Processor
Acc = byte.
CRC calculator & save byte.
next byte.
end of block
branch if it isn't.

```
F852 20 7B F8     (.    JSR  &F87B   } SAVE CRCs for Data block
F855 20 84 F8     ..    JSR  &F884   }
F858 20 84 F8     ..    JSR  &F884   }
F85B 20 46 FB     F.    JSR  &FB46   reset 6850
F85E A9 01        ..    LDA  #&01    } short delay (2 x 5 = 10 frames)
F860 20 98 F8     ..    JSR  &F898   }
F863 28           (     PLP          recover JSR
F864 20 B9 F8     ..    JSR  &F8B9   output file info to screen.
F867 2C CA 03     ,..   BIT  &03CA   block flag byte.
F86A 10 08        ..    BPL  &F874   if not last block branch (exit)
F86C 08           .     PHP          save JSR
F86D 20 92 F8     ..    JSR  &F892   long delay.
F870 20 46 F2     F.    JSR  &F246
F873 28           (     PLP          recover JSR.
F874 60           `     RTS          exit

F875 20 82 F8     ..    JSR  &F882
F878 4C B0 F7     L..   JMP  &F7B0   (CRC calculator)

F87B A5 BF        ..    LDA  &BF     } SAVE CRC bytes to TAPE (if ? &BD = byte.)
F87D 20 82 F8     ..    JSR  &F882   }
F880 A5 BE        ..    LDA  &BE     }
F882 85 BD        ..    STA  &BD

F884 20 95 F9     ..    JSR  &F995   CFS timeout flag test & check if Escape pressed!
F887 24 C0        $.    BIT  &C0
F889 10 F9        ..    BPL  &F884
F88B A9 00        ..    LDA  #&00    ? &C0 = 0
F88D 85 C0        ..    STA  &C0
F88F A5 BD        ..    LDA  &BD     A = byte from cassette (?!)
F891 60           `     RTS          (only relevant during load.

F892 A9 32        .2    LDA  #&32
F894 D0 02        ..    BNE  &F898   (always taken)

F896 A5 C7        ..    LDA  &C7
F898 A2 05        ..    LDX  #&05
F89A 8D 40 02     .@.   STA  &0240   CFS timeout counter
F89D 20 95 F9     ..    JSR  &F995   wait for timeout? / Escape test
F8A0 2C 40 02     ,@.   BIT  &0240   CFS timeout counter
F8A3 10 F8        ..    BPL  &F89D
F8A5 CA           .     DEX
F8A6 D0 F2        ..    BNE  &F89A
F8A8 60           `     RTS

F8A9 AD C6 03     ...   LDA  &03C6   } Branch if Block = 0
F8AC 0D C7 03     ...   ORA  &03C7   }
F8AF F0 05        ..    BEQ  &F8B6   }
F8B1 2C DF 03     ,..   BIT  &03DF   } branch if not last block.
F8B4 10 03        ..    BPL  &F8B9   }
F8B6 20 49 F2     I.    JSR  &F249

F8B9 A0 00        ..    LDY  #&00    &BA = 0
F8BB 84 BA        ..    STY  &BA
F8BD AD CA 03     ...   LDA  &03CA   block flag byte
F8C0 8D DF 03     ...   STA  &03DF   copy of block flag.
F8C3 20 DC E7     ..    JSR  &E7DC   ?
F8C6 F0 6B        .k    BEQ  &F933   ? exit
F8C8 A9 0D        ..    LDA  #&0D    } CR
F8CA 20 EE FF     ..    JSR  OSWRCH  }
F8CD B9 B2 03     ...   LDA  &03B2,Y File name
F8D0 F0 10        ..    BEQ  &F8E2   end of file name.
F8D2 C9 20        .     CMP  #&20    } ASC of character < space (32)
F8D4 90 04        ..    BCC  &F8DA   }
F8D6 C9 7F        ..    CMP  #&7F    } " " " > & 7F (delete).
F8D8 90 02        ..    BCC  &F8DC   }
F8DA A9 3F        .?    LDA  #&3F    - Yes output a '?' if character < 32 or > &7E
F8DC 20 EE FF     ..    JSR  OSWRCH  output character
F8DF C8           .     INY          next character
```

```
F8E0 D0 EB       ..      BNE   &F8CD
F8E2 AD 47 02    .G.     LDA   &0247
F8E5 F0 04       ..      BEQ   &F8EB
F8E7 24 BB       $.      BIT   &BB
F8E9 50 48       PH      BVC   &F933
F8EB 20 91 F9     ..     JSR   &F991
F8EE C8          .       INY
F8EF C0 0B       ..      CPY   #&0B
F8F1 90 EF       ..      BCC   &F8E2
F8F3 AD C6 03    ...     LDA   &03C6
F8F6 AA          .       TAX
F8F7 20 7A F9     z.     JSR   &F97A
F8FA 2C CA 03    ,..     BIT   &03CA
F8FD 10 34       .4      BPL   &F933
F8FF 8A          .       TXA
F900 18          .       CLC
F901 6D C9 03    m..     ADC   &03C9
F904 85 CD       ..      STA   &CD
F906 20 75 F9     u.     JSR   &F975
F909 AD C8 03    ...     LDA   &03C8
F90C 85 CC       ..      STA   &CC
F90E 20 7A F9     z.     JSR   &F97A
F911 24 BB       $.      BIT   &BB
F913 50 1E       P.      BVC   &F933
F915 A2 04       ..      LDX   #&04
F917 20 91 F9     ..     JSR   &F991
F91A CA          .       DEX
F91B D0 FA       ..      BNE   &F917
F91D A2 0F       ..      LDX   #&0F
F91F 20 27 F9     '.     JSR   &F927
F922 20 91 F9     ..     JSR   &F991
F925 A2 13       ..      LDX   #&13
F927 A0 04       ..      LDY   #&04
F929 BD B2 03    ...     LDA   &03B2,X
F92C 20 7A F9     z.     JSR   &F97A
F92F CA          .       DEX
F930 88          .       DEY
F931 D0 F6       ..      BNE   &F929
F933 60          `       RTS
F934 AD 47 02    .G.     LDA   &0247
F937 F0 03       ..      BEQ   &F93C
F939 4C 10 E3    L..     JMP   &E310
F93C 20 8E FB     ..     JSR   &FB8E
F93F 20 E2 FB     ..     JSR   &FBE2
F942 20 DC E7     ..     JSR   &E7DC
F945 F0 EC       ..      BEQ   &F933
F947 20 46 FA     F.     JSR   &FA46
F94A 52          R       ???
F94B 45 43       EC      EOR   &43
F94D 4F          O       ???
F94E 52          R       ???
F94F 44          D       ???
F950 20 74 68     th     JSR   &6874
F953 65 6E       en      ADC   &6E
F955 20 52 45     RE     JSR   &4552
F958 54          T       ???
F959 55 52       UR      EOR   &52,X
F95B 4E 00 20    N.      LSR   &2000
F95E 95 F9       ..      STA   &F9,X
F960 20 E0 FF     ..     JSR   OSRDCH
F963 C9 0D       ..      CMP   #&0D
F965 D0 F6       ..      BNE   &F95D
```

TAPE/ROM Flag
if tape branch
} output a space.
block low
X= block number.
output block number to screen
** block flag byte
branch if not the last block (exit)
length of block (high) — &CD
length of block (low) — &CC
exit if bit 6 of &BB is 0. (set by *OP
output a space. (4 spaces).
} Print load address.
output a space.
} setup offsets for execution address
used to output load & execution address of a file.
read cassette/ROM FS flag.
branch if tape selected
if ROM selected issue 'Bad Command'
cassette relay on (*MOTOR) 1
reset 6850 & set up serial ULA for operation
??
exit. (is this a fail?).
Print following message.
'RECORD then RETURN' not error message.
JSR &F995
} wait for 'Return'.

```
F967 4C E7 FF    L..    JMP  OSNEWL
F96A E6 B1       ..     INC  &B1
F96C D0 06       ..     BNE  &F974
F96E E6 B2       ..     INC  &B2
F970 D0 02       ..     BNE  &F974
F972 E6 B3       ..     INC  &B3
F974 60          `      RTS
F975 48          H      PHA
F976 20 91 F9     ..    JSR  &F991
F979 68          h      PLA
F97A 48          H      PHA
F97B 4A          J      LSR  A
F97C 4A          J      LSR  A
F97D 4A          J      LSR  A
F97E 4A          J      LSR  A
F97F 20 83 F9     ..    JSR  &F983
F982 68          h      PLA
F983 18          .      CLC
F984 29 0F       ).     AND  #&0F
F986 69 30       iO     ADC  #&30
F988 C9 3A       .:     CMP  #&3A
F98A 90 02       ..     BCC  &F98E
F98C 69 06       i.     ADC  #&06
F98E 4C EE FF    L..    JMP  OSWRCH
F991 A9 20       .      LDA  #&20
F993 D0 F9       ..     BNE  &F98E
F995 08          .      PHP
F996 24 EB       $.     BIT  &EB
F998 30 04       0.     BMI  &F99E
F99A 24 FF       $.     BIT  &FF
F99C 30 02       0.     BMI  &F9A0
F99E 28          (      PLP
F99F 60          `      RTS
F9A0 20 3B F3     ;.    JSR  &F33B
F9A3 20 F2 FA     ..    JSR  &FAF2
F9A6 A9 7E       .~     LDA  #&7E
F9A8 20 F4 FF     ..    JSR  OSBYTE
F9AB 00          .      BRK
F9AC 11 45       .E     ORA  (&45),Y
F9AE 73          s      ???
F9AF 63          c      ???
F9B0 61 70       ap     ADC  (&70,X)
F9B2 65 00       e.     ADC  &00
F9B4 98          .      TYA
F9B5 F0 0D       ..     BEQ  &F9C4
F9B7 20 46 FA     F.    JSR  &FA46
F9BA 0D 4C 6F    .Lo    ORA  &6F4C
F9BD 61 64       ad     ADC  (&64,X)
F9BF 69 6E       in     ADC  #&6E
F9C1 67          g      ???
F9C2 0D 00 85    ...    ORA  &8500
F9C5 BA          .      TSX
F9C6 A2 FF       ..     LDX  #&FF
F9C8 A5 C1       ..     LDA  &C1
F9CA D0 0D       ..     BNE  &F9D9
F9CC 20 72 FA     r.    JSR  &FA72
F9CF 08          .      PHP
F9D0 A2 FF       ..     LDX  #&FF
F9D2 A0 99       ..     LDY  #&99
F9D4 A9 FA       ..     LDA  #&FA
F9D6 28          (      PLP
F9D7 D0 1C       ..     BNE  &F9F5
```

next line.

('cassette critical flag'
if bit 7 set then no output? P276 ULA
test escape
branch if Escape pressed
exit

'Escape routine'

'Escape' &11

'Loading' not error message

STA &BA.

```
F9D9 A0 8E        ..    LDY  #&8E
F9DB A5 C1        ..    LDA  &C1
F9DD F0 04        ..    BEQ  &F9E3
F9DF A9 FA        ..    LDA  #&FA
F9E1 D0 12        ..    BNE  &F9F5
F9E3 AD C6 03     ...   LDA  &03C6
F9E6 C5 B4        ..    CMP  &B4
F9E8 D0 07        ..    BNE  &F9F1
F9EA AD C7 03     ...   LDA  &03C7
F9ED C5 B5        ..    CMP  &B5
F9EF F0 13        ..    BEQ  &FA04
F9F1 A0 A4        ..    LDY  #&A4
F9F3 A9 FA        ..    LDA  #&FA
F9F5 48           H     PHA
F9F6 98           .     TYA
F9F7 48           H     PHA
F9F8 8A           .     TXA
F9F9 48           H     PHA
F9FA 20 B6 F8      ..   JSR  &F8B6
F9FD 68           h     PLA
F9FE AA           .     TAX
F9FF 68           h     PLA
FA00 A8           .     TAY
FA01 68           h     PLA
FA02 D0 14        ..    BNE  &FA18
FA04 8A           .     TXA
FA05 48           H     PHA
FA06 20 A9 F8      ..   JSR  &F8A9
FA09 20 D6 FA      ..   JSR  &FAD6
FA0C 68           h     PLA
FA0D AA           .     TAX
FA0E A5 BE        ..    LDA  &BE
FA10 05 BF        ..    ORA  &BF
FA12 F0 79        .y    BEQ  &FA8D
FA14 A0 8E        ..    LDY  #&8E
FA16 A9 FA        ..    LDA  #&FA
FA18 C6 BA        ..    DEC  &BA
FA1A 48           H     PHA
FA1B 24 EB        $.    BIT  &EB
FA1D 30 0D        0.    BMI  &FA2C
FA1F 8A           .     TXA
FA20 2D 47 02     -G.   AND  &0247
FA23 D0 07        ..    BNE  &FA2C
FA25 8A           .     TXA
FA26 29 11        ).    AND  #&11
FA28 25 BB        %.    AND  &BB
FA2A F0 10        ..    BEQ  &FA3C
FA2C 68           h     PLA
FA2D 85 B9        ..    STA  &B9
FA2F 84 B8        ..    STY  &B8
FA31 20 8B F6      ..   JSR  &F68B
FA34 46 EB        F.    LSR  &EB
FA36 20 E8 FA      ..   JSR  &FAE8
FA39 6C B8 00     l..   JMP  (&00B8)
FA3C 68           h     PLA
FA3D C8           .     INY
FA3E D0 03        ..    BNE  &FA43
FA40 18           .     CLC
FA41 69 01        i.    ADC  #&01
FA43 48           H     PHA
FA44 98           .     TYA
FA45 48           H     PHA
```

Always taken

Checks if block load is correct.

Y = &A4
A = &FA

Save A
Save Y, X

Print Filename/blocknumber etc.

Recover X, Y, A

exit if CRCs....

Branch if OPT 2,0 selected(?)

```
FA46 20 DC E7     ..    JSR  &E7DC
FA49 A8           .     TAY
FA4A 68           h     PLA
FA4B 85 B8        ..    STA  &B8
FA4D 68           h     PLA
FA4E 85 B9        ..    STA  &B9
FA50 98           .     TYA
FA51 08           .     PHP
FA52 E6 B8        ..    INC  &B8
FA54 D0 02        ..    BNE  &FA58
FA56 E6 B9        ..    INC  &B9
FA58 A0 00        ..    LDY  #&00
FA5A B1 B8        ..    LDA  (&B8),Y
FA5C F0 0A        ..    BEQ  &FA68
FA5E 28           (     PLP
FA5F 08           .     PHP
FA60 F0 F0        ..    BEQ  &FA52
FA62 20 E3 FF      ..   JSR  OSASCI
FA65 4C 52 FA     LR.   JMP  &FA52
FA68 28           (     PLP
FA69 E6 B8        ..    INC  &B8
FA6B D0 02        ..    BNE  &FA6F
FA6D E6 B9        ..    INC  &B9
FA6F 6C B8 00     l..   JMP  (&00B8)
FA72 A2 FF        ..    LDX  #&FF
FA74 E8           .     INX
FA75 BD D2 03     ...   LDA  &03D2,X
FA78 D0 07        ..    BNE  &FA81
FA7A 8A           .     TXA
FA7B F0 03        ..    BEQ  &FA80
FA7D BD B2 03     ...   LDA  &03B2,X
FA80 60           `     RTS
FA81 20 E3 E4      ..   JSR  &E4E3
FA84 5D B2 03     ]..   EOR  &03B2,X
FA87 B0 02        ..    BCS  &FA8B
FA89 29 DF        ).    AND  #&DF
FA8B F0 E7        ..    BEQ  &FA74
FA8D 60           `     RTS
FA8E 00           .     BRK
FA8F D8           .     CLD
FA90 0D 44 61     .Da   ORA  &6144
FA93 74           t     ???
FA94 61 3F        a?    ADC  (&3F,X)
FA96 00           .     BRK
FA97 D0 15        ..    BNE  &FAAE
FA99 00           .     BRK
FA9A DB           .     ???
FA9B 0D 46 69     .Fi   ORA  &6946
FA9E 6C 65 3F     le?   JMP  (&3F65)
FAA1 00           .     BRK
FAA2 D0 0A        ..    BNE  &FAAE
FAA4 00           .     BRK
FAA5 DA           .     ???
FAA6 0D 42 6C     .Bl   ORA  &6C42
FAA9 6F           o     ???
FAAA 63           c     ???
FAAB 6B           k     ???
FAAC 3F           ?     ???
FAAD 00           .     BRK
FAAE A5 BA        ..    LDA  &BA
FAB0 F0 21        .!    BEQ  &FAD3
FAB2 8A           .     TXA
```

wait for cassette

Print text following return address.

exit

Print message until BRK (0).

return to program at end of text

File being searched for

branch if not end of Filename.

A = offset for 0 byte at end of filename
if 0 then it's looking for a null file.

A to lower case filename just found

to lower case.

'Data?' &D8

'File?' &DB

'Block?' &DA

```
FAB3 F0 1E       ..     BEQ  &FAD3
FAB5 A9 22       ."     LDA  #&22
FAB7 24 BB       $.     BIT  &BB
FAB9 F0 18       ..     BEQ  &FAD3
FABB 20 46 FB     F.    JSR  &FB46
FABE A8          .      TAY
FABF 20 4A FA     J.    JSR  &FA4A
FAC2 0D 07 52    ..R    ORA  &5207
FAC5 65 77       ew     ADC  &77
FAC7 69 6E       in     ADC  #&6E   (beep 'Rewind tape' not error.
FAC9 64          d      ???
FACA 20 74 61     ta    JSR  &6174
FACD 70 65       pe     BVS  &FB34
FACF 0D 0D 00    ...    ORA  &000D
FAD2 60          `      RTS
FAD3 20 4D F2     M.    JSR  &F24D
FAD6 A5 C2       ..     LDA  &C2
FAD8 F0 F8       ..     BEQ  &FAD2
FADA 20 95 F9     ..    JSR  &F995
FADD AD 47 02    .G.    LDA  &0247
FAE0 F0 F4       ..     BEQ  &FAD6
FAE2 20 88 F5     ..    JSR  &F588
FAE5 4C D6 FA    L..    JMP  &FAD6
FAE8 20 DC E7     ..    JSR  &E7DC  check cassette control block.
FAEB F0 05       ..     BEQ  &FAF2
FAED A9 07       ..     LDA  #&07   } beep.
FAEF 20 EE FF     ..    JSR  OSWRCH }
FAF2 A9 80       ..     LDA  #&80
FAF4 20 BD FB     ..    JSR  &FBBD  test for tube - tube call A=&
FAF7 A2 00       ..     LDX  #&00   } relay off.
FAF9 20 95 FB     ..    JSR  &FB95  }
FAFC 08          .      PHP
FAFD 78          x      SEI
FAFE AD 82 02    ...    LDA  &0282  Copy of serial ULA
FB01 8D 10 FE    ...    STA  &FE10  serial ULA control reg.
FB04 A9 00       ..     LDA  #&00   } RS423 timeout = 0.
FB06 85 EA       ..     STA  &EA    }
FB08 F0 01       ..     BEQ  &FB0B
FB0A 08          .      PHP
FB0B 20 46 FB     F.    JSR  &FB46  reset 6850.
FB0E AD 50 02    .P.    LDA  &0250  RS423 control block
FB11 4C 89 E1    L..    JMP  &E189  write to 6850 control register.
FB14 28          (      PLP         retrieve status
FB15 24 FF       $.     BIT  &FF    test escape flag.
FB17 10 18       ..     BPL  &FB31  if escape not pressed branch (repeat lo
FB19 60          `      RTS         exit
FB1A A5 E3       ..     LDA  &E3
FB1C 0A          .      ASL  A
FB1D 0A          .      ASL  A
FB1E 0A          .      ASL  A
FB1F 0A          .      ASL  A
FB20 85 BB       ..     STA  &BB
FB22 AD D1 03    ...    LDA  &03D1
FB25 D0 08       ..     BNE  &FB2F
FB27 A5 E3       ..     LDA  &E3    cassette FS options (*OPT),
FB29 29 F0       ).     AND  #&F0   clear bits 0-3 (leave load/save options
FB2B 85 BB       ..     STA  &BB    store at &BB.
FB2D A9 06       ..     LDA  #&06   } &C7=6 ?.
FB2F 85 C7       ..     STA  &C7    }
FB31 58          X      CLI  IR on
FB32 08          .      PHP  save status
FB33 78          x      SEI  IR off
```

```
FB34 2C 4F 02     ,O.    BIT  &024F      RS423 use flag.
FB37 10 DB        ..     BPL  &FB14
FB39 A5 EA        ..     LDA  &EA        RS423 timeout counter
FB3B 30 D7        0.     BMI  &FB14
FB3D A9 01        ..     LDA  #&01       } RS423 timeout counter=1= CFS using 6850
FB3F 85 EA        ..     STA  &EA        }
FB41 20 46 FB      F.    JSR  &FB46      . reset 6850.
FB44 28           (      PLP             retrieve
FB45 60           `      RTS             exit.
FB46 A9 03        ..     LDA  #&03       } 3 into &FE08 = Master Reset.
FB48 D0 1B        ..     BNE  &FB65      }
FB4A A9 30        .0     LDA  #&30
FB4C 85 CA        ..     STA  &CA
FB4E D0 13        ..     BNE  &FB63
FB50 A9 05        ..     LDA  #&05
FB52 8D 10 FE     ...    STA  &FE10
FB55 A2 FF        ..     LDX  #&FF       } delay.
FB57 CA           .      DEX             }
FB58 D0 FD        ..     BNE  &FB57      }
FB5A 86 CA        ..     STX  &CA
FB5C A9 85        ..     LDA  #&85
FB5E 8D 10 FE     ...    STA  &FE10
FB61 A9 D0        ..     LDA  #&D0
FB63 05 C6        ..     ORA  &C6
FB65 8D 08 FE     ...    STA  &FE08      6850 control register.
FB68 60           `      RTS
FB69 AE C6 03     ...    LDX  &03C6      } current block number
FB6C AC C7 03     ...    LDY  &03C7      }
FB6F E8           .      INX             Inc current block and store in &B4,&B5
FB70 86 B4        ..     STX  &B4        ∴ &B4,B5 = next block expected.
FB72 D0 01        ..     BNE  &FB75
FB74 C8           .      INY
FB75 84 B5        ..     STY  &B5
FB77 60           `      RTS
FB78 A0 00        ..     LDY  #&00
FB7A 84 C0        ..     STY  &C0
FB7C 84 BE        ..     STY  &BE
FB7E 84 BF        ..     STY  &BF
FB80 60           `      RTS
FB81 A0 FF        ..     LDY  #&FF
FB83 C8           .      INY
FB84 E8           .      INX
FB85 BD 00 03     ...    LDA  &0300,X
FB88 99 D2 03     ...    STA  &03D2,Y
FB8B D0 F6        ..     BNE  &FB83
FB8D 60           `      RTS
FB8E A0 00        ..     LDY  #&00
FB90 58           X      CLI
FB91 A2 01        ..     LDX  #&01
FB93 84 C3        ..     STY  &C3        &C3=0
FB95 A9 89        ..     LDA  #&89       *FX137,1,0   relay on, for write operations
FB97 A4 C3        ..     LDY  &C3                    (AUG-P161)
FB99 4C F4 FF     L..    JMP  OSBYTE     - exit.
FB9C 85 BC        ..     STA  &BC
FB9E 98           .      TYA
FB9F 4D 47 02     MG.    EOR  &0247
FBA2 A8           .      TAY
FBA3 A5 E2        ..     LDA  &E2
FBA5 25 BC        %.     AND  &BC
FBA7 4A           J      LSR  A
FBA8 88           .      DEY
FBA9 F0 04        ..     BEQ  &FBAF
```

```
FBAB 4A          J      LSR  A
FBAC 88          .      DEY
FBAD D0 02       ..     BNE  &FBB1
FBAF B0 4D       .M     BCS  &FBFE
FBB1 00          .      BRK
FBB2 DE 43 68    .Ch    DEC  &6843,X
FBB5 61 6E       an     ADC  (&6E,X)
FBB7 6E 65 6C    nel    ROR  &6C65
FBBA 00          .      BRK
FBBB A9 01       ..     LDA  #&01
FBBD 20 D3 FB     ..    JSR  &FBD3
FBC0 F0 3C       .<     BEQ  &FBFE
FBC2 8A          .      TXA
FBC3 A2 B0       ..     LDX  #&B0
FBC5 A0 00       ..     LDY  #&00
FBC7 48          H      PHA
FBC8 A9 C0       ..     LDA  #&C0
FBCA 20 06 04     ..    JSR  &0406
FBCD 90 FB       ..     BCC  &FBCA
FBCF 68          h      PLA
FBD0 4C 06 04    L..    JMP  &0406
FBD3 AA          .      TAX
FBD4 A5 B2       ..     LDA  &B2
FBD6 25 B3       %.     AND  &B3
FBD8 C9 FF       ..     CMP  #&FF
FBDA F0 05       ..     BEQ  &FBE1
FBDC AD 7A 02    .z.    LDA  &027A
FBDF 29 80       ).     AND  #&80
FBE1 60          `      RTS
FBE2 A9 85       ..     LDA  #&85
FBE4 8D 10 FE    ...    STA  &FE10
FBE7 20 46 FB     F.    JSR  &FB46
FBEA A9 10       ..     LDA  #&10
FBEC 20 63 FB     c.    JSR  &FB63
FBEF 20 95 F9     ..    JSR  &F995
FBF2 AD 08 FE    ...    LDA  &FE08
FBF5 29 02       ).     AND  #&02
FBF7 F0 F6       ..     BEQ  &FBEF
FBF9 A9 AA       ..     LDA  #&AA
FBFB 8D 09 FE    ...    STA  &FE09
FBFE 60          `      RTS
FBFF 00          .      BRK
FC00 FF          .      ???
FC01 FF          .      ???
FC02 FF          .      ???
FC03 FF          .      ???
FC04 FF          .      ???
FC05 FF          .      ???
FC06 FF          .      ???
FC07 FF          .      ???
FC08 FF          .      ???
FC09 FF          .      ???
FC0A FF          .      ???
FC0B FF          .      ???
FC0C FF          .      ???
FC0D FF          .      ???
FC0E FF          .      ???
FC0F FF          .      ???
FC10 FF          .      ???
FC11 FF          .      ???
FC12 FF          .      ???
FC13 FF          .      ???
```

'Channel' &DE

A=1
Test for tube.
— exit if tube isn't present.

Acc
Put A on stack.
} claim tube ?.
Acc = 0 (always) ?
enter tube code at &406.

&B2 } 2
&B3 } 6

— exit
tube flag

leave bit 7 as it is & remove bits 0-6.

turn 'Motor on led' on & select 300 baud f
serial ULA register. TX.
reset the 6850.
(or A location &C6 & store result in 6850 co
check cassette critical flag. regis
6850 status register.
test bit
is it 0. & if so branch (this means TDR
transmit an &AA is Fu
(6850 Transmit data register)
TDR.

exit

End of OS 1.20. #1

```
*D FF00 FFFF
FF00 20 51 FF      Q.     JSR  &FF51
FF03 20 51 FF      Q.     JSR  &FF51
FF06 20 51 FF      Q.     JSR  &FF51
FF09 20 51 FF      Q.     JSR  &FF51
FF0C 20 51 FF      Q.     JSR  &FF51
FF0F 20 51 FF      Q.     JSR  &FF51
FF12 20 51 FF      Q.     JSR  &FF51
FF15 20 51 FF      Q.     JSR  &FF51
FF18 20 51 FF      Q.     JSR  &FF51
FF1B 20 51 FF      Q.     JSR  &FF51
FF1E 20 51 FF      Q.     JSR  &FF51
FF21 20 51 FF      Q.     JSR  &FF51
FF24 20 51 FF      Q.     JSR  &FF51
FF27 20 51 FF      Q.     JSR  &FF51
FF2A 20 51 FF      Q.     JSR  &FF51
FF2D 20 51 FF      Q.     JSR  &FF51
FF30 20 51 FF      Q.     JSR  &FF51
FF33 20 51 FF      Q.     JSR  &FF51
FF36 20 51 FF      Q.     JSR  &FF51
FF39 20 51 FF      Q.     JSR  &FF51
FF3C 20 51 FF      Q.     JSR  &FF51
FF3F 20 51 FF      Q.     JSR  &FF51
FF42 20 51 FF      Q.     JSR  &FF51
FF45 20 51 FF      Q.     JSR  &FF51
FF48 20 51 FF      Q.     JSR  &FF51
FF4B 20 51 FF      Q.     JSR  &FF51
FF4E 20 51 FF      Q.     JSR  &FF51

FF51 48            H      PHA
FF52 48            H      PHA
FF53 48            H      PHA
FF54 48            H      PHA
FF55 48            H      PHA
FF56 08            .      PHP
FF57 48            H      PHA
FF58 8A            .      TXA
FF59 48            H      PHA
FF5A 98            .      TYA
FF5B 48            H      PHA
FF5C BA            .      TSX
FF5D A9 FF         ..     LDA  #&FF
FF5F 9D 08 01      ...    STA  &0108,X
FF62 A9 88         ..     LDA  #&88
FF64 9D 07 01      ...    STA  &0107,X
FF67 BC 0A 01      ...    LDY  &010A,X
FF6A B9 9D 0D      ...    LDA  &0D9D,Y
FF6D 9D 05 01      ...    STA  &0105,X
FF70 B9 9E 0D      ...    LDA  &0D9E,Y
FF73 9D 06 01      ...    STA  &0106,X
FF76 A5 F4         ..     LDA  &F4
FF78 9D 09 01      ...    STA  &0109,X
FF7B B9 9F 0D      ...    LDA  &0D9F,Y
FF7E 85 F4         ..     STA  &F4
FF80 8D 30 FE      .0.    STA  &FE30
FF83 68            h      PLA
FF84 A8            .      TAY
FF85 68            h      PLA
FF86 AA            .      TAX
FF87 68            h      PLA
FF88 40            @      RTI
FF89 08            .      PHP
```

&2200 2 4 6 8 A C E
&2210 2 4 6 8 A C E
&2220 2 4 6 8 A C E
&2230 2 4

Expanded Vector calls.

For Paged ROMS

OLD ROM socket goes to &FF89 - remember it holds the address +1, on the stack

&FF88 } ROM address of call.
1
2
3 — low
4 — high
5

SR — 6
A — 7
X — 8
Y — 9
X=SP

2 = &FF

3 = &88

Y = Low of original JSR read from table (low)

5 = from vector table read from table (high).

4 = from vecto table

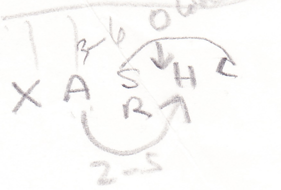
1 = previous ROM (present ROM).

select ROM from vector table. Make OS copy.

restore Y
" X
" A
remove SR & call the routine.

Save SR

Original &FFXX ← call. Still on Stack



```
FF8A 48          H      PHA  Save A
FF8B 8A          .      TXA } Save X
FF8C 48          H      PHA }
FF8D BA          .      TSX   X=SP.
FF8E BD 02 01    ...    LDA  &0102,X } copy A to high of
FF91 9D 05 01    ...    STA  &0105,X
FF94 BD 03 01    ...    LDA  &0103,X } copy SR to low of
FF97 9D 06 01    ...    STA  &0106,X
FF9A 68          h      PLA } restore X
FF9B AA          .      TAX
FF9C 68          h      PLA  – "  A
FF9D 68          h      PLA  — SR,
FF9E 68          h      PLA      OLD ROM socket.
FF9F 85 F4       ..     STA  &F4
FFA1 8D 30 FE    .0.    STA  &FE30
FFA4 68          h      PLA  A  ?
FFA5 28          (      PLP  SR
FFA6 60          `      RTS  exit from ROM call.   OSGBPB entry + other other Invalid vecto
FFA7 8A          .      TXA                        *FX 157   Fast TUBE  BPUT.
FFA8 B0 2A       .*     BCS  &FFD4
FFAA BC 00 FC    ...    LDY  &FC00,X               *FX 146   Read  FRED.
FFAD 60          `      RTS
FFAE BC 00 FD    ...    LDY  &FD00,X               *FX 148   Read  JIM.
FFB1 60          `      RTS
FFB2 BC 00 FE    ...    LDY  &FE00,X               *FX 150   Read  SHEILA.
FFB5 60          `      RTS
FFB6 36 40       6@     ROL  &40,X
FFB8 D9 4C 0B    .L.    CMP  &0B4C,Y RDRM  JMP &DC0B
FFBB DC          .      ???
FFBC 4C C0 C4    L..    JMP  &C4C0  Main VDU character output entry point.
FFBF 4C 94 E4    L..    JMP  &E494  EVEN
FFC2 4C 1E EA    L..    JMP  &EA1E  GSINIT
FFC5 4C 2F EA    L/.    JMP  &EA2F  GS READ
FFC8 4C C5 DE    L..    JMP  &DEC5  WRCH (Non vectored)
FFCB 4C A4 E0    L..    JMP  &E0A4  RDCH (  "      "  )
FFCE 6C 1C 02    l..    JMP  (&021C) FIND
FFD1 6C 1A 02    l..    JMP  (&021A) GBPB
FFD4 6C 18 02    l..    JMP  (&0218) BPUT
FFD7 6C 16 02    l..    JMP  (&0216) BGET
FFDA 6C 14 02    l..    JMP  (&0214) ARGS
FFDD 6C 12 02    l..    JMP  (&0212) FILE
FFE0 6C 10 02    l..    JMP  (&0210) RDCH
FFE3 C9 0D       ..     CMP  #&0D    ASCI
FFE5 D0 07       ..     BNE  &FFEE
FFE7 A9 0A       ..     LDA  #&0A    NEWL
FFE9 20 EE FF     ..    JSR  OSWRCH
FFEC A9 0D       ..     LDA  #&0D
FFEE 6C 0E 02    l..    JMP  (&020E) WRCH
FFF1 6C 0C 02    l..    JMP  (&020C) WORD
FFF4 6C 0A 02    l..    JMP  (&020A) BYTE
FFF7 6C 08 02    l..    JMP  (&0208) CLI
FFFA 00          .      BRK          NMI   &D00
FFFB 0D CD D9    ...    ORA  &D9CD   RESET &D9CD
FFFE 1C          .      ???          BRK   &DC1C - interrupt requests 'IRQ'.
FFFF DC          .      ???
0000 02          .      ???
0001 19 02 19    ...    ORA  &1902,Y
?OFF
```

End of OS 1.20.